Energy Interoperation Common Transactive Services (CTS) Version 1.0

Committee Specification Draft 05

Deleted: 04

17 February 2025

Deleted: 09 September 2024¶

This stage:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd05/ei-cts-v1.0-csd05.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd05/ei-cts-v1.0-csd05.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd05/ei-cts-v1.0-csd05.docx

Previous stage:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd04/ei-cts-v1.0-csd04.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd04/ei-cts-v1.0-csd04.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd04/ei-cts-v1.0-csd04.docx

Latest stage:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.docx

Deleted: Previous stage:¶
(Authoritative)¶

Technical Committee:
OASIS Energy Interoperation TC

Chairs:
William T. Cox (wtcox@coxsoftwarearchitects.com), Individual
David Holmberg (david.holmberg@nist.gov), NIST

Editor:
Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill

Related work:

This document replaces or supersedes:

- Common Transactive Services 1.0. The Energy Mashup Lab Specification. Edited by William T. Cox, Toby Considine 30 November 2020. https://www.theenergymashuplab.org/s/cts-1-0-draft-20201130.pdf.

This document is related to:

- *Energy Interoperation Version 1.0*. Edited by Toby Considine, 11 June 2014. OASIS Standard. http://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html. Latest version: http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html. and its TeMIX Profile
- *Energy Market Information Exchange (EMIX) Version 1.0.* Edited by Toby Considine. Latest version: http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html.
- *WS-Calendar Platform Independent Model (PIM) Version 1.0*. Edited by William Cox and Toby Considine. Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html.

- *Schedule Signals and Streams Version 1.0*. Edited by Toby Considine and William T. Cox. Latest version: http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html.

Abstract:

Common Transactive Services (CTS) permits energy consumers and producers to interact through energy markets by simplifying actor interaction with any market. CTS is a streamlined and simplified profile of the OASIS Energy Interoperation (EI) specification, which describes an information and communication model to coordinate the exchange of energy between any two Parties that consume or supply energy, such as energy suppliers and customers, markets and service providers.

Status

This document was last revised or approved by the OASIS Energy Interoperation TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://groups.oasis-open.org/communities/tc-community-home2?CommunityKey=d3a22f3f-887d-465e-bb0f-018dc7d3f405#technical.

Deleted: Statusx¶

Comments from TC members should be sent directly to the TC's mailing list. Comments may be submitted to the project by any other person through the use of the project's Comment Facility: https://groups.oasis-open.org/communities/community-home?communitykey=70a647c6-d0e6-434c-8b30-018dce25fd35

This document is provided under the RF on Limited Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this document, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/energyinterop/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Key words:

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

Citation format:

When referencing this document, use the following citation format:

[Energyinterop-CTS-v1.0]

Energy Interoperation Common Transactive Services (CTS) Version 1.0. Edited by Toby Considine. 09 September 2024. OASIS Committee Specification Draft 04. https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd04/ei-cts-v1.0-csd04.html. Latest stage: https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html.

Notices:

Deleted: 2024

Table of Contents

Table of Tables

Table of Figures

1 Introduction

Transactive energy names the collaboration techniques to balance energy supply and energy demand at every moment even as power generation becomes decentralized and as the ownership of energy assets becomes more diverse. The OASIS Energy Interoperation 1.0 ([EI]) specification defined the interactions and communication required for transactive energy.

The Common Transactive Services (CTS) is an application profile of [EI] with most optionality and complexity stripped away. CTS is strongly influenced by both the TEMIX profile of [EI] and by the philosophy behind TEMIX. CTS defines the messages for transactive energy, leaving communication details unspecified. CTS extends the TEMIX approach using lessons learned in the world's largest financial markets. CTS is both a simplification and extension of [EI] and not part of EI.

The purpose of CTS is to enable broad semantic interoperation between systems in transactive energy-based markets, or in any markets whose products are commodities distinguished chiefly by time of delivery. These time-volatile commodities are termed resources, and the interactions defined in CTS are common to any market used to manage resources over time.

To encourage broad adoption, CTS uses terms from financial markets in preference to the relatively obscure terms used in specialized energy markets. The Technical Committee has collaborated through a liaison with the FIX Trading Community[1], whose specifications are supported by essentially all global financial markets.

Among these is the use of the term instrument for a tradeable asset, or a negotiable item. In CTS, the term instrument encompasses a quantity of a Resource delivered at a particular time for a particular duration. A transaction is created when a buyer and seller agree on the price for an instrument.

Transactive resource markets coordinate resource supply and resource use through markets that trade instruments. The initial research into transactive resource markets used a market to allocate heat from a single furnace within a commercial building. Transactive resource markets balance supply and demand over time using automated voluntary transactions between market participants.

Examples of transactable resources include, but are not limited to, electrical energy, electrical power, natural gas, and thermal energy such as steam, hot water, or chilled water. The capability to transmit such time-dependent resources is also a transactable resource, as instruments can be defined for transmission rights as well as for the services that maintain grid frequency or voltage.

When we apply transactive resource markets to the distribution of power or energy, we refer to it as transactive energy. A significant driver of transactive energy is the desire to smooth supply and demand variability, or alternatively, to match demand to variable supply. We anticipate this variability to increase as additional variable and distributed generation sources are connected to the power grid. The reader can find an extended discussion of Transactive Energy (TE) in the EI specification [EI]

A goal of CTS is to enable systems and devices developed today or in the future to address the challenges of increasing distributed energy resources. CTS enables distributed actors to participate in markets deployed today or in the future.

CTS defines interactions between actors in energy markets. We do not identify whether an actor is a single integrated system, or a distributed collection of systems and devices working together. See Section 1.6 for a discussion of the term Actor in this specification. Autonomous market actors must be able to recognize patterns and make choices to best support their own needs.

CTS assumes the perspective of a *trader*, that is of a market participant. [EI] was developed with significant input form Economists and energy market regulators, and it relies on language from economics and regulation. The Committee deliberately chose to seek guidance from financial traders and to use their language. Many data elements and message types have been renamed to align with FIX-based financial markets.

[1] https://www.fixtrading.org/

CTS messages are simple and strongly typed and make no assumptions about the systems or technologies behind the actors. Rather, CTS defines a technology-agnostic minimal set of messages to enable interoperation through markets of participants irrespective of internal technology. In a similar manner, CTS does not specify the internal organization of a market, but rather a common set of messages that can be used to communicate with any transactive energy market.

The Common Transactive Services, strictly speaking, are a definition of the payloads and exchange patterns necessary for a full-service environment for interaction with markets. In other words, CTS describes the message payloads to be exchanged, defining the semantic content and ordering of messages. Any message exchange mechanism may be used, including but not limited to message queues and Service-Oriented mechanisms.

In a Service-Oriented Architecture [SOA] environment, the semantic payloads are those sent and returned by the *services* described. CTS enables any SOA or other framework to exchange equivalent semantic information without presuming the specific messaging system(s) or architecture used, thus allowing straightforward semantic interoperation.[2] See Section 2.3.

1.1 Application of the Common Transactive Services

The purpose of this specification is to codify the common interactions and messages required for energy markets. Any system able to use CTS should be able to interoperate with any CTS-conforming market with minimal or no change to system logic. The full protocol stack and cybersecurity requirements for message exchange between systems using CTS are out of scope.

Systems that can be represented by CTS actors include but are not limited to:

- Smart Buildings/Homes/Industrial Facilities
- Building systems/devices
- Business Enterprises
- Electric Vehicles
- Microgrids
- Collections of IoT (Internet of Things) devices

TE demonstrations and deployments have seldom been interoperable—each uses its own message model and its own market dynamics. Systems built to participate in these demonstrations and deployments are not able to interoperate with other implementations. The intent of this specification is to enable systems and markets developed for future deployments to interoperate even as the software and markets continue to evolve.

CTS does not presume a Market with a single seller (e.g., a utility). CTS recognizes two parties to a transaction, and the role of any Party can switch from buyer to seller from one transaction to the next. Each Resource Offer (Tender) has a Side attribute (Buy or Sell). When each transaction is committed (once the product has been purchased), it is owned by the purchaser, and it can be re-sold as desired or needed.

A CTS micromarket may balance power over time in a traditional distribution system attached to a larger power grid or it may bind to and operate a stand-alone autonomous microgrid **[SmartGridBusiness].**

1.2 Support for Developers

Specific coding, message, and protocol recommendations are beyond the scope of this specification which specifies information content and interactions between systems. The Common Transactive Services payloads are described using the Universal Modelling Language [UML]. Many software development tools can accept artifacts in UML or in XSD to enforce proper message formation.

The Committee plans to release artifacts defining the commonly used XML and JSON schemas.

[2] SOA is occasionally mis-described as a *client-server* approach. In distinction, services are requested by an Actor, and fulfilled by another Actor. In SOA the services offered are key, and the actors take different roles in different interactions.

The FIX Simple Binary Encoding (SBE)[3] is used in financial markets and for general high-performance messaging—SBE is designed to encode and decode messages using fewer CPU instructions than standard encodings and without forcing memory management delays. SBE-based messaging is used when very high rates of message throughput are required. The TC plans to release a SBE schema as well.

All Schemas will be in a separate release after this specification is complete.

1.3 Naming Conventions

This specification follows some naming conventions for artifacts defined by the specification, as follows:

For the names of elements and the names of attributes within XSD files and UML models, the names follow the lowerCamelCase convention, with all names starting with a lower-case letter. For example,

```
<element name="componentType" type="ei:ComponentType"/>
```

For the names of types within XSD files, the names follow the UpperCamelCase convention with all names starting with an upper-case letter suffixed by "type-". For example,

```
<complexType name="ComponentServiceType">
```

For clarity in UML models the suffix "type" is not always used.

For the names of intents and for attributes in the UML models, names follow the lowerCamelCase convention, with all names starting with a lower-case letter, EXCEPT for cases where the intent represents an established acronym, in which case the entire name is in upper case.

JSON and where possible SBE names follow the same conventions.

1.4 Editing Conventions

For readability, element names in tables appear as separate words. Actual names are lowerCamelCase, as specified above, and as they appear in the UML models, and in the XML and JSON schemas.

All elements in the tables not marked as "optional" are mandatory. This is the opposite of the convention used in the specification of FIX Protocol.

Information in the **FIX Field** column is non-normative and includes in parentheses zero or more FIX Tags that are related to the field. This provides guidance for those integrating CTS markets to interoperate with markets supporting the FIX Protocol.

Information in the **Meaning** column of the tables is normative. Information appearing in the **Notes** column is explanatory and non-normative.[4]

Examples and Appendices are non-normative. In particular, architectural and functional examples are presented only to support narrative description. The specific processes, structures, and algorithms are out of scope.

1.5 FIX and the Language of Trading

As noted above, this specification strives to apply the language of financial trading to resource markets. FIX is the language of trading.

We thank members of the FIX Trading Community (https://www.fixtrading.org/) for their extensive input and close reading. FIX was formed in 1991 to connect the global ecosystem of venues, asset managers,

[3] SBE has been submitted to ISO/IEC JTC1 as a standard (see https://www.iso.org/standard/90107.html) where it is currently in the ballot stage.

[4] In ISO and IEC standards, portions that are not normative are *informative*. OASIS uses the term *non-normative*.

banks/brokers, vendors and regulators by standardizing the communication among participants. FIX relies on 4 key principles:

- Creating and maintaining robust open standards across the across the trade life-cycle with its pre-trade, trade, and post-trade environments.
- Providing advice and counsel to regulatory bodies in a transparent and unbiased way.
- Seeking ways to improve the trading process front to back for the global financial services industry.
- Providing FIX members with a neutral, collaborative environment to come together through member-driven committees, working groups and conferences to promote, support and educate.

This specification relied strongly on their assistance.

1.6 Use of terms Actors and Facets in this specification

This specification defines message content and interaction patterns.

The EI 1.0 specification in 2011 presumed web services for interactions. That specification described a Service-Oriented Architecture (SOA) approach. Service orientation complements loose integration and organizes distributed capabilities that may be in different ownership domains by focusing solely on requested results rather than on mechanisms. [EI] uses the language of web services to describe all interactions.

There is a growing use of the descriptive term “cloud-native computing” for extending the architecture and technologies developed for use in clouds not only in data centers but to edge computing, where IoT devices reside. A discussion of the rapidly evolving topics of cloud-native computing and edge computing is beyond the scope of this specification.

At the time of this specification, typical architectures decompose applications into smaller, independent building blocks that are easier to develop, deploy and maintain. A single market participant in energy may be embodied as several of these independent blocks (actors).

For the Internet of Things (IoT), the term Actor begins and ends at the interfaces to things. The “actor model” makes no assumptions of the mechanisms or even motives internal to an Actor. An Actor is simply a thing that acts. The Actor may be instantiated by software in a traditional computer, a cloud node, by a human behind a user interface, or by a device on the Internet of things.

In transactive energy, the actor model supports the diversity of IoT and of markets. An energy seller may be a generator or a solar panel or a virtual power plant or a demand responsive facility or a financial entity. An energy buyer or seller may be a home or commercial facility or an embedded device or a microgrid or an energy district. A Market acts to match Tenders. An Actor may take a market-maker role, buying and/or selling power for itself. An energy storage system may act as a buyer or as a seller at any time.

We use the term “Facet” to name a cohesive set of messages that an Actor may use to communicate with other Actors. An Actor submits tenders to buy or to sell. An Actor may operate a Market. If the Architecture includes a telemetry Actor, measuring Resource flow (metering), then that Actor MAY represent the Market or the market participant or even a third party. This specification makes no requirement as to how to distribute or make use of these facets.

While this specification discusses messages between Actors, it establishes no requirement or expectation of specific implementation. While this specification uses the language of Actor and Facet, there is no architectural expectation linked to this language. One could apply the terms Actor and Facet throughout the [EI] specification. A traditional [EI] application consisting of several unitary systems each presenting all facets as web services described by WSDL can be conformant so long as it uses a compatible set of information payloads.

1.7 Security and Privacy

Service requests and responses are generally considered public actions of each interoperating system, with limitations to address privacy and security considerations (see Appendix C). Service actions are independent from private actions behind the interface (i.e., device control actions). A Facet is used without needing to know all the details of its implementation. Consumers of services generally pay for results, not for effort.

1.7.1 Security Considerations

Size of transactions, costs of failure to perform, confidentiality agreements, information stewardship, and even changing regulatory requirements can require that similar transactions be expressed within quite different security contexts. Loose integration using the service-oriented architecture (SOA) style assumes careful definition of security requirements between partners. It is a feature of the SOA approach that security is composed in order to meet the specific and evolving needs of different markets and transactions. Security implementation is free to evolve over time and to support different needs. The Common Transactive Services allow for this composition, without prescribing any particular security implementation.

The best practice in cloud-native computing is to use Zero Trust security **[ZeroTrust]**. Zero Trust security requires authentication and authorization of every device, person, and application. The best practice is to encrypt all messages, even those between the separate components of an application within the cloud.

This specification makes no attempt to describe methods or technologies to enable Zero Trust interactions between Actors.

1.7.2 Privacy Considerations

Detailed knowledge of offers to buy or sell or knowledge of energy inputs and outputs for an Actor may reveal information on actions and operations. For example, transactions or tenders may indicate whether a production line is starting or stopping, or anticipated energy needs, or who has been buying or selling power. Making such information public may be damaging to actors. Similarly, an adverse party may be able to determine the likelihood that a dwelling is presently occupied.

The essence of any transaction is the agreement of a Party to sell, and of another Party to buy. The identity of the buyer and the identity of the seller are each part of the transaction. Some transaction notifications may hide the identity of the buyer from the seller. Some transaction notifications may hide the identity of the seller from the buyer. Some transactions, such as those arising from what the energy world calls a double auction[5], may be between the market participants as a whole, and not with any particular counterparty. Where required, the Market itself may be designated as the counterparty in a notification.

Both security and privacy considerations are addressed in Appendix C.

1.8 Semantic Composition

The semantics and interactions of CTS are selected from and derived from OASIS Energy Interoperation [EI]. EI references two other standards, [EMIX] and [WS-Calendar], and uses an earlier Streams definition. We adapt, update, and simplify the use of the referenced standards, while maintaining conformance.

- Appendix D Semantic Composition from and relationship to Energy Interoperation, EMIX, and WS-Calendar describes price and product for electricity markets. WS-Calendar communicates schedules and sequences of operations.
- EI uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each EI service references a product defined using [EMIX]. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EI] is expressed using [WS-Calendar].
- Since OASIS published [EI], a semantically equivalent but simpler [Streams] specification was developed in the OASIS WS-Calendar Technical Committee. CTS uses that simpler [Streams] specification.

[5] In a double auction, there are tenders to buy and tenders to sell, and all participants clear at the same price. FIX simply uses the term “Auction”.

See Appendix D, Semantic Composition from and relationship to Energy Interoperation, EMIX, and WS-Calendar.

In [EI], the fundamental resource definition was the [EMIX] Item, composed of a resource name, a unit of measure, a scale factor, and a quantity. For example, a specific EMIX Item may define a Market denominated in 25 MWh bids. In CTS, we group and name these elements as a Resource, Product, and Instrument. These terms are defined in Section 2.2.4, “Markets and Market Segments”

Note that the informational elements in a fully defined tender or transaction are identical to those described in EMIX. The conceptual regrouping enables common behaviors including Market discovery and interoperation between Actors built on different code bases.

1.9 Applicability to Microgrids (Informative)

As an extended example, using the Common Transactive Services terminology, a microgrid is comprised of interacting nodes each represented by an actor (interacting as CTS parties). Those actors interact in a micromarket co-extensive in scope with the microgrid. No actor reveals any internal mechanisms, but only its interest in buying and selling power.

An actor can represent a microgrid within a larger micromarket; the actor would in effect aggregate the resources in the microgrid. As above, such an actor would not reveal any internal mechanisms, but only its interest in buying and selling power. There is no explicit bound on repeating this interoperation pattern.

An actor representing a microgrid may interoperate with markets in a regional grid, which may or may not be using CTS. In addition, infrastructure capacity may limit delivery to the microgrid. The Actor representing a microgrid must translate and enforce constraints and share information with the other nodes in the microgrid solely by means of CTS. Any translations or calculations performed are out of scope.

See informative references [StructuredEnergy] and [SmartGridBusiness] for a discussion. [Fractal Microgrids] is an early reference that describes hierarchies of microgrids. [Transactive Microgrids] describes transactive energy in microgrids.

1.10 Specific scope statements

This specification interprets Energy Interoperation from the perspective of a Trader interacting with a Market. CTS defines Pre-Trade, Trade, and Post-Trade information exchanges. Trading refers to the specific interactions that buy or sell a resource. A Trader uses pre-trade information to discern the operation of the Market and the actions of the other Traders. Post-Trade information informs the participants of the Trade, tracks whether the resource is delivered, and any resulting changes to the Trader’s ability to participate in the Market.

Interaction patterns and facet definitions to support the following are in scope for Common Transactive Services:

- Interaction patterns to support transactive energy, including tenders, transactions, and supporting information.
- Information models for price and Product communication.
- Information models for Market and Market Segment characteristics.
- Payload definitions for Common Transactive Services.

The following are out of scope for Common Transactive Services:

- Requirements specifying the type of agreement, contract, Product definition, or tariff used by a particular market.
- Computations or agreements that describe how power is sold into or sold out of a market.
- Communication protocols, although semantic interaction patterns are in scope.

This specification describes standard messages, the set of which may be extended.

1.11 Naming of Messages and Operations

The naming of messages and operations and message payloads follows the pattern defined in [EI]. Services are named starting with the letters ***Ei*** following the Upper Camel Case convention. Operations in each service use one or more of the following patterns. The first listed is a fragment of the name of the initial service operation; the second is a fragment of the name of the response message which acknowledges receipt, describes errors, and may pass information back to the invoker of the first operation.

Create—Created	An object is created and sent to the other Party.
Cancel—Canceled	A previously created request is canceled.

For example, to construct an operation name for the Tender Facet, "Ei" is concatenated with the name fragment (verb) as listed. An operation to cancel an outstanding Tender is called *EiCancelTender*.[6]

Facets describe what would be called services in a full Service-Oriented Architecture implementation, as we do not define SOA services, but only imply and follow a service structure from [EI].

[6] This pattern was developed and is used by IEC Technical Committee 57 (Power Systems).

2 Overview of Common Transactive Services

CTS provides for the exchange of resources among actors, in the role of parties, which represent any provider or consumer of energy. Systems use CTS to interoperate in transactive resource markets. A transactive resource market balances the supply of a resource over time and the demand for that resource by using a market specifying the time of delivery.

Although the Common Transactive Services are a profile and extension of Energy Interoperation, the CTS focus is markets and trading. The language used in the Energy Interoperation specification was developed with extensive input from economists, regulators, and participants in highly regulated markets.

CTS strives to use the language of financial markets and traders.

This specification supports agreements and transactional obligations, while offering flexibility of implementation to support specific approaches and goals of the various participants.

2.1 Parties

This CTS specification defines defines interactions between participants in a resource market. This Resource Market is a means to make collaborative decisions that allocate power or other resources over time. We follow [EI] and financial markets by calling market participants “Parties”.

When the market recognizes tenders that match each other, however decided, the market generates a transaction that represents a contract (“Trade”) between the buyer and the seller. This transaction includes a party and a counterparty.

2.2 Trading semantics from FIX Protocol

The FIX Protocol divides messages into Pre-Trade, Trade, and Post-Trade Messages.

Deleted: Community

Pre-Trade messages convey information that traders need to discover how to use the market and to develop a strategy to buy and sell successfully. Pre-Trade messages include market data (“Tickers”) of bids, offers and contracts in the market (“Orders” and “Quotes”). Other Pre-Trade messages provide reference data, describing how the market itself works and what a Party can expect when interacting with the market.

Deleted: trade

Deleted: advertisements and announcements…

Deleted: ,

Deleted: negotiations between parties (“…

Deleted: pre

Deleted: describe

Trade messages include submitting and cancelling orders (“Tenders”) to the market and executing contracts (“Transactions”) when orders to sell match (however defined) orders to buy.

Post-trade messages in FIX include allocation, confirmation, settlement, position and collateral management. CTS does not include allocation or collateral management.

For narrative purposes, this specification begins with the Trade facets: Tenders and Transactions. It then discusses the post-trade facets of Delivery and Position. This covers all the functions in some transactive resource markets. This specification then describes Negotiation, an optional Pre-Trade facet. It next describes the Pre-Trade market data (“Tickers”) that inform an Actor about the activities of other participants. The Pre-Trade Instrument Market Data facet provides summary information about Tenders currently held in the market. Finally, the Pre-Trade Market Structure facet conveys how a Trader may interact with the market, which includes how to find each facet and which messages this market supports.

Deleted: pre

Deleted: pre-trade

Deleted: reports

Deleted: pre-trade

Deleted: pre-trade

Deleted: Facet

An Actor interacting with the market would first discover the market structure, subscribe to Tickers relevant to its interest, and then use the facets and messages that are permitted in this market to Trade. A Party MAY not understand Negotiation, or MAY skip subscribing to Tickers, but any party MUST be able to Trade.

Deleted: negotiation

When available, this specification references matching field names, tag numbers, and values from the FIX Protocol. FIX Protocol field names are upper camel case and we follow their convention that the field name is followed by the tag number in parentheses, as in FieldName(0).

Deleted: and field

Deleted: fieldnames

Deleted: generally

Deleted: field

2.2.1 Parties and Orders

In Energy Interop as in FIX, a trade is executed between two parties. While Energy Interoperation acknowledges only a Party and a Counterparty, FIX is more semantically rich.

What Energy Interoperation (and this specification) terms Tenders, FIX terms orders. An order that is on the book in the market is a Resting or Passive order. An order that enters the market to match a Resting order is the Initiating or Aggressive order. Passive orders increase market liquidity. Aggressive orders decrease market liquidity when they match to existing orders. Regulators of financial markets are often interested in liquidity and in the ratios of Aggressive to Passive orders.

When it makes the discussion clearer, this specification uses the terms Resting, Passive, Initiating, and Aggressive as they are used in financial markets.

2.2.2 Instruments

Financial Markets trade financial instruments. CTS borrows this language from FIX. See Section 3, Market Semantics: Resource, Product, Instrument, for a discussion.

2.2.3 Market Crossing

Market Crossing refers to either the opening or to the closing of a market or market segment. A traditional exchange opens in the morning and closes in the afternoon. Tenders are not matched prior to market opening or after the market close.

In many markets, parties wishing to trade pay close attention to prices and volumes in the period around closing. Many traders prefer not to trade close to a crossing because it is a period of high price volatility on a market. Many markets announce an indicative “closing price” and an indicative “opening price”, even though no transaction may occur at either of those prices. The actual opening or closing price is determined as first step of the uncrossing of a market, which ends the crossing phase.

Deleted: have occurred

As transactive resource markets are in essence markets in time of delivery, individual instruments can be considered to open and close as well. In a continuously open market segment, a rule might prevent trading more than 24 hours in advance. In that same market, an instrument for delivery of a resource between 10:00 AM and 11:00 AM may no longer be traded at noon in the previous day.

Transactive resource markets may have regulatory time limits on trading. Some electricity markets have banned transactions more than a day prior to delivery. CTS traders must be able to understand the local rules and adjust their trading tactics without human intervention. A Market MAY accept Tenders prior to the opening of the Market Segment or Instrument. Transactive market researchers have used tenders submitted prior to opening to generate opening prices in black-start scenarios. Others have used trade residue, which is the tenders left in the market after closing to seed real-time prices for unplanned energy use.

2.2.4 Markets and Market Segments

Systems use the Common Transactive Services to interoperate in transactive resource markets. A transactive resource market balances the supply of a resource over time and the demand for that resource by using a market and instruments specifying the time of delivery.

Deleted: common transactive services…

A Market is composed of different segments wherein different products are traded, perhaps with different rules. The Market and all its segments trade a single Resource. Following the FIX Protocol, we term these Market Segments, and we use the FIX Market Model Typology (MMT[7]) to name the market activities (Market Mechanism and Trading Mode) of each Segment. A Market may have one or many Market Segments.

Deleted: classification

Deleted: Mechanism Type

[7] The MMT standard (https://www.fixtrading.org/mmt/) originated from an initiative of the Federation of European Securities Exchanges aiming at improving the consistency and comparability of data from different data sources. In order for the MMT standard to become more widely recognized and adopted, MMT has been placed under the FIX Protocol Limited Trust.

Deleted: Since then, MMT has been developed together with a broader range of industry participants, including trading venues, data vendors and data users, and has expanded into new asset classes.…

2.3 Common Transactive Services Roles

Actors interact through messages submitted to Facets. The specification makes no assertions about the behaviors, processes, or motives within each Actor. A particular Actor may use all Facets, a subset of Facets, or even a single Facet. This specification groups similar messages by Facet messages and interactions.

2.3.1 Parties as Market Participants

The Common Transactive Services (CTS) defines interactions in a Resource Market. This Resource Market is a means to make collaborative decisions that allocate power or other Resource over time. We follow [EI] and financial markets by calling market participants “Parties”.

A Party can take one of two Sides in Transaction:

- Buy, or
- Sell

A Party selling an Instrument takes the Sell Side of the Transaction. A Party buying an Instrument takes the Buy Side of the Transaction. The initiating Party is called the Party in a Transaction; the other Party is called the Counterparty.

From the perspective of the Market, there is no distinction between a Party selling additional power and Party selling from its previously acquired position. An Actor representing a generator would generally take the Sell side of a transaction. An Actor representing a consumer generally takes the Buy side of a transaction. However, a generator may take the Buy Side of a Transaction to reduce its own generation, in response either to changes in physical or market conditions or to reflect other commitments made by the Actor. A consumer may choose to sell from its current position if its plans change, or if it receives an attractive price. A power storage system actor may choose to buy or sell from Interval to Interval, consistent with its operating and financial goals.

Deleted: actor

We do not specify how to manage delivery of the Resource.

2.3.2 Party and Counterparty and Transactions

The party in a tender is offering to buy or sell. The PartyID in a Tender should always reference the Party that is tendering.

When the Market recognizes tenders that *match* each other (however defined), the market generates a Transaction that represents an agreement between the buyer and the seller. This Transaction includes sending a EiTransaction message to both the Party and a Counterparty. If the match was composed from multiple Tenders, each party receives an EiTransaction for each Tender matched.

Deleted: transaction

Deleted: transaction

Deleted: Party

2.3.3 Facets in the CTS Specification

This specification refers to a cohesive set of interactions, that is, closely related requests and responses, as Facets. A Party sends and receives defined messages through one or more Facets. A Party may be composed of one or more Actors, each with one or more Facets. A Party may communicate with its composite Actors through the same Facets or through other Facets not defined in this specification.

Actors use Facets to interact with other Actors that expose a complementary Facet. An Actor in a CTS-based system of systems may expose all Facets, a single Facet, or any collection of Facets. A particular Market may use some or all named Facets. A participant in a Market must include Actors supporting each Facet required in that Market; there is no requirement that each Actor supports all these Facets.

Detailed descriptions of each facet begin in Section 4.

Table 2-1: Facets Defined in CTS

Facet	Description
Registration	A Party must Register with a Market to participate in the Market Segments in that Market. See Section 4, "Party Registration Facet".
Tender	Tenders are actionable offers to buy or to sell an Instrument at a given price. Tenders may be sent to a specific counterparty or sent to the whole Market Segment, published via a Ticker to all Parties in the Market Segment. See Section 5, "The Tender Facet".
Transaction	A Transaction records the trade when a Tender to buy and a Tender to sell are matched. Each Party is notified of the creation of the Transaction. Note: a Tender for one side MAY match more than one Tender on the other side, and could generate multiple Transactions, potentially at different prices. See Section 6, "The Transaction Facet".
Position	At any moment, a Party has a position which represents the cumulative quantity for each of the Instruments that the Party has previously transacted for within a bounding time interval across all Segments in the Market. A Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold. Note that parties that can store or generate power or that can buy from another market MAY be able to sell more than their market position. See Section 7, "The Position Facet".
Delivery	It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed. Some implementations may compare what was purchased or sold with what was delivered. What a system does after this comparison is out of scope. See Section 8, "The Delivery Facet".
Negotiation	Negotiation uses messages that may lead to a Tender that will be accepted. Negotiation includes Requests for Quotes (RFQs), Quotes, and Quote Responses. See Section 9, "The Negotiation Facet".
Tickers	A Ticker is a continuous live view of market interactions—consider the historical ticker tape. A Ticker is one form of Market Subscriptions as defined by FIX. See Section 11, "Tickers"
Market Instrument Summaries	A Market Instrument Summary is a compressed or summarized variant of Market Data as defined by FIX. See Section 12 "Instrument Data Subscriptions"
Market Reference and Dynamic Data	The Reference Data Facet communicates Market Reference Data that describes the Market and each Market Segment; Session data is more dynamic. An Actor may query the Market to discover the Resource and Products traded in a Market. While a Market trades a single Resource, it may consist of multiple Market Segments trading multiple Products. See Section 13 "Market Structure Reference Data: Market, Segment, and Session Subscriptions"

Each of these facets includes multiple messages which are described starting in Section 0 below. Sometimes the use of one facet precedes the use of another facet, as Tenders may initiate messages that result in messages for the Transaction Facet.

2.4 Responses

This section re-iterates terms, simplifies, and extends models from [EI]. The form of the Response is common across all Facets.

Figure 2-1 shows the UML class diagram for responses and the Response Detail enumeratiioin.

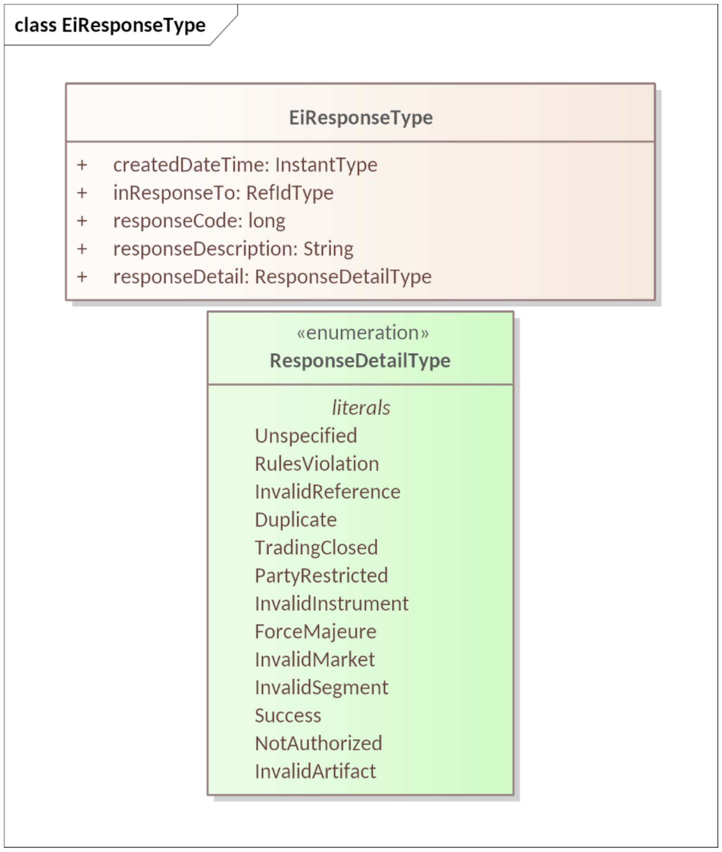


Figure 2-1 UML Class Diagram of EiResponseType

Attributes for responses are shown in Table 2-2. The various attribute types are not in FIX.

Table 2-2: Attributes of EiResponse

Attribute	Type	Meaning
Created DateTime	Instant Type	Timestamp for creation of this response
In Response To	Ref ID Type	A reference ID which identifies the artifact or message element to which this is a response. The Request ID uniquely identifies this request and can serve as a messaging correlation ID[8].

[8] As an example of the *Correlation Pattern* for messages

Deleted: The

Deleted:

Deleted: and MarketAttributeViolationType…

Deleted: Date Time

Deleted: Market Attribute Violation
Pairs of strings ... [1]

Attribute	Type	Meaning
Response Code	Long	The Response Code indicates success or failure of the operation requested. The Response Description is unconstrained text, perhaps for use in a user interface. The code ranges are those used for HTTP response codes,[9] specifically: 1xx: Informational - Request received, continuing process. 2xx: Success - The action was successfully received, understood, and accepted 3xx: Pending - Further action must be taken in order to complete the request 4xx: Requester Error - The request contains bad syntax or cannot be fulfilled 5xx: Responder Error - The responder failed to fulfill an apparently valid request Response codes for the. Facets are described in the respective sections. For the Tender Facet see Section 5.5.
Response Detail	Response Deetail Type	An enumeration that gives more detail on the response reason. See table below.
Response Description	String	A string describing the response, e.g. “Duration doesn’t match Segment configured Duration”

Deleted: Most

Many messages elicit a response. Information-only messages, as in Tickers, do not. The enumeration liiterals for Response Detail are shown in Table 2-3.

Table 2-3 Enumeration Response Detail Type

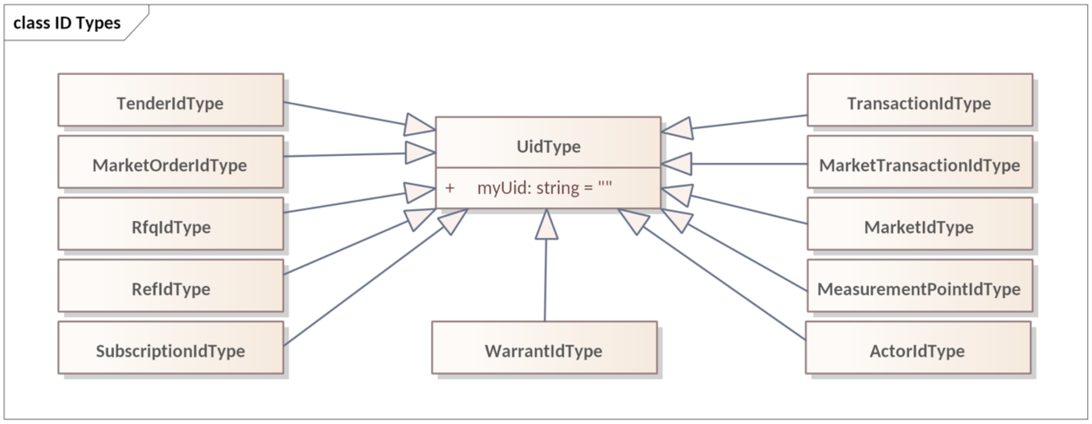
Literal	Meaning
Success	The action was successfully received, understood, and accepted
Unspecified	An unspecified error occurred.
Duplicate	The operation submitted a duplicate artifact.
Force Majeure	The trading venue of its own accord rejected the operation. A similar situation arises when a market cancels a tender to maintain market integrity or liquiiity.
Invalid Artifact	The artifact (Tender, Transaction, Quote, RFQ) included is not valid.
Invalid Instrument	The instrument specified is invalid or is otherwise not tradeable. Reasons include (a) the instrument is outside the Segment’s Tradeable Interval, the Duration does not match that for the product traded in the target segment, or Parties are not resolvable.
Invalid Market	The MarketID is not valid
Invalid Reference	A referenced object (e.g. Referenced Quote ID in an EiAcceptQuote Payload)
Invalid Segment	The Segment ID is not valid in the Market specified
Not Authorized	The party invoking the operation is not authorized
Party Restricted	One of the parties is not presently permitted to trade.
Rules Violation	Attributes from Segment, Market, or Session data are violated. For example, lot size, maximum order size, and the like.

[9] See e.g. https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Literal	Meaning
Trading Closed	The targeted segment was not open for trading when the request was received. Hours are in the respective data.

2.5 Identities

In general, CTS uses specific types that inherit from UID Type, with a string as the inherited attribute. This allows representation of unique identifiers variously called UIDs, GUIDs, and other names, while maintaining type safety.



Deleted:

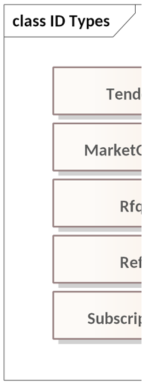


Figure 2-2 UML Class Diagram of ID Types in CTS

3 Market Semantics: Resource, Product, Instrument, and Streams

The messages of CTS use a few common elements. These elements derive from and are compatible with definitions in [WS-Calendar], [EMIX], and in [EI].

Every CTS-based market offers the exchange of a specific resource. Each CTS market segment is a venue for trading a single product, which is a resource packaged for sale. All tenders and transactions are for instruments, which are products scheduled for delivery at a specific time.

3.1 Resource, Product, & Instrument

We define a Resource as a commodity whose value depends on time of delivery. A Party subscribes (see Section 10) to a Market to discover the Resource that is traded in the market, and the Products available in different Market Segments. (See Section 13 “Market Structure Reference Data: Market, Segment, and Session Subscriptions”) A Party can then trade Instruments, a Product at a specific time, in a Market Segment. This specification leaves Market Definition until the end of the specification, as the meaning and import of the terms used to define each Segment are first described in the trading process.

Figure 3-1 illustrates the relationship between Resource, Product and Instrument. This is expressed formally as UML in Figure 3-2. The relationship is illustrated twice, with an informal sketch and with formal UML below.

Understanding these three terms is essential to understanding CTS.

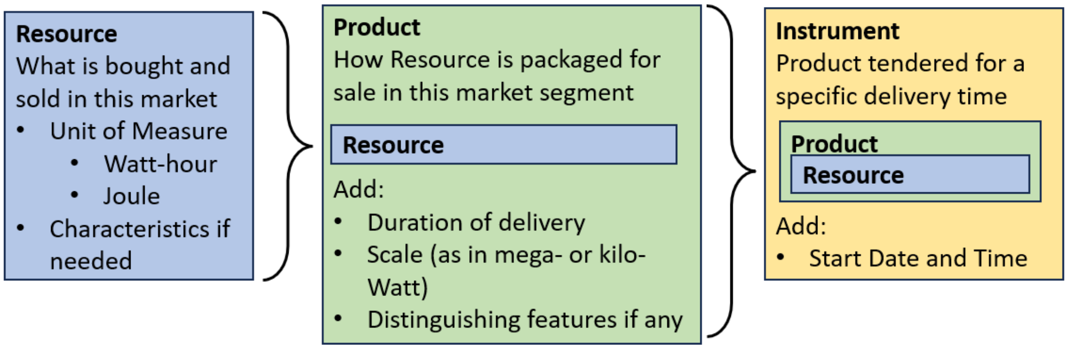

Resource
What is bought and sold in this market
• Unit of Measure
• Watt-hour
• Joule
• Characteristics if needed
Product
How Resource is packaged for sale in this market segment
Resource
Add:
• Duration of delivery
• Scale (as in mega- or kilo-Watt)
• Distinguishing features if any
Instrument
Product tendered for a specific delivery time
Product
Resource
Add:
• Start Date and Time

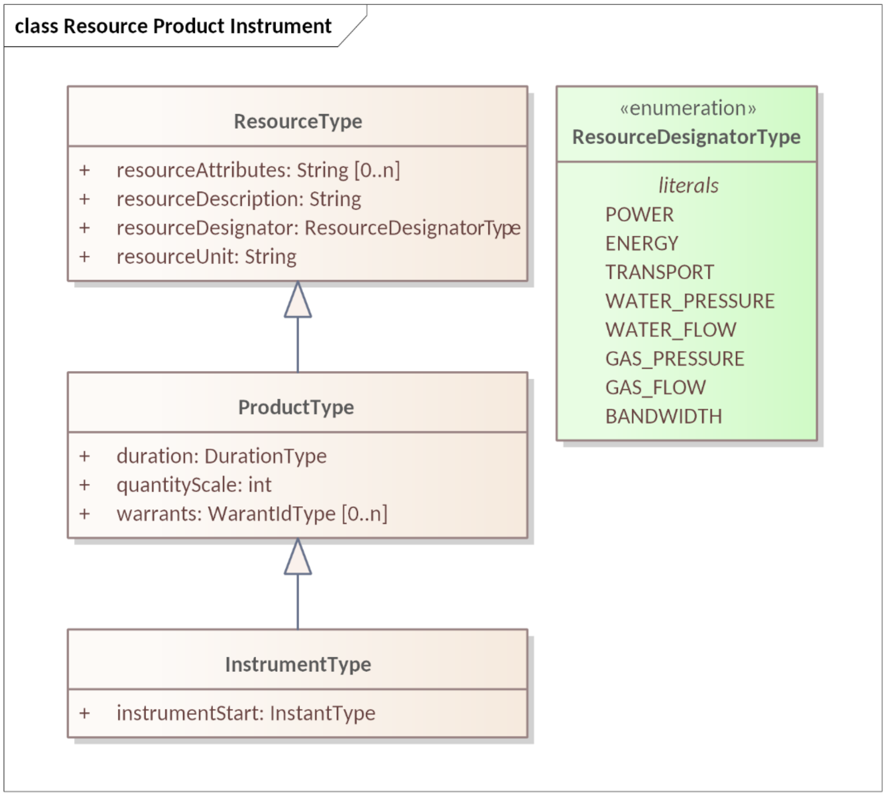
Figure 3-1 Informal sketch showing relationship between Resource, Product, and Instrument

The Product incorporates the Resource, defining how the Resource is “packaged” for market. Adding a start date-time to a Product defines an Instrument.

A Market Segment trades Instruments, as a financial market trades financial instruments. CTS trades Instruments to deliver Product at a specific time.

A Market trades a single Resource; a Market Segment trades a sinigle Product.

The UML in Figure 3-2 shows the relationship between Resource, Product, and Instrument.



Deleted:

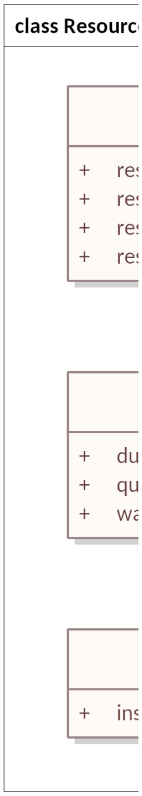


Figure 3-2 UML Class Diagram for Resource, Product, and Instrument

3.1.1 Defining Resource

We define a Resource as a commodity whose value depends on time of delivery. A developer may extend the Resource enumeration using standard UML techniques (subclassing); however, CTS 1.0 uses only the limited list in the Resource Designator Type (Figure 3-2).

A Market typically includes some information that further specifies the Resource, for example voltage and frequency for Power.

Table 3-1: Defining the Resource

Attribute	Type	FIX Field	Meaning	Notes
Resource Attributes	String	Not in FIX	Optional elements that further describe the Resource	e.g. Hertz and Voltage. Different Commodities will require different attributes to be specific.
Resource Description	String	Not in FIX	Text description of the Resource	

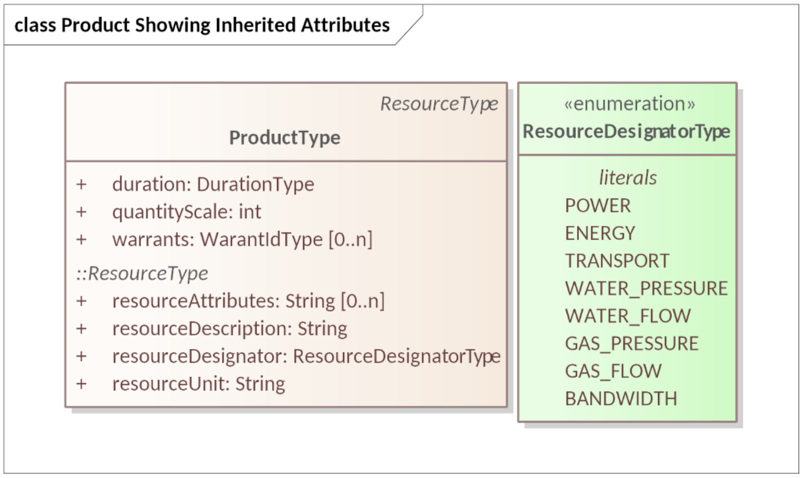
Attribute	Type	FIX Field	Meaning	Notes
Resource Designator	String	Not in FIX	POWER ENERGY TRANSPORT WATER_PRESSURE WATER_FLOW GAS_PRESSURE GAS_FLOW BANDWIDTH	The Resource Designator serves a purpose similar to that of the FIX AssetSubClass(1939) with AssetClass(1938)=5 (Commodity) The list is extensible
Resource Unit	String	Not in FIX	The unit of measure for the Resource	Item Unit in [EMIX] The Resource Unit serves a purpose similar to that of the FIX UnitOfMeasure (996)

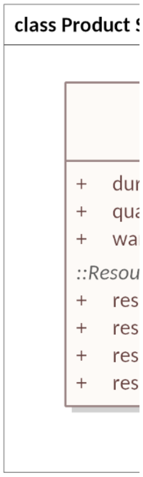
The Resource is named in the Market. Each Market deals in a single Resource. Segments of a Market restrict trading into profiles of the Resource. Position and Delivery (see Sections 7, 8 below) itemize Resource quantities.

3.1.2 Defining Product

The Product is a Resource packaged for Market. The size and duration of the Product define what is, in effect, the “package size” for the commodity. A Market may offer multiple Products for the same Resource in different Market Segments.

Note that the Product is derived from the [EMIX] ItemBase.





Deleted:

Figure 3-3 UML Class Diagram for Product showing Inheritance from Resource

Table 3-2, below, defines each of the fields in the Product.

Table 3-2: Defining the Product

Attribute	Type	FIX Field	Meaning	Notes
Duration	Duration Type	Not in FIX	The interval Duration for the specific Product definition.	As defined in **[WS-Calendar]**

Attribute	Type	FIX Field	Meaning	Notes
Quantity Scale	Integer	Not in FIX	A scale factor for Resource Units; the number of resource units in a trade of quantity one of an instrument.	Example 1: A Product measured in kilowatts where the resource uniit is Watts has a Quantity Scale of 1000. Example 2: In a Segment with a Quantity Scale of 1000, a trade of one unit is a trade of one thousand of the Resource Unit—if the resource unit is Watt-hours and Quantity Scale 1000, a trade of quantity 1 is a trade of one kWatt hour. (kWh) FIX expresses this with UnitOfMeasure equal to “kWh” and UnitOfMeasureQty equal to 1.
Warrants (Optional)	Warrant ID Type	Not in FIX	Optional further specificity of Product.	Warrants that MAY be available are itemized in the Market. This specification does not define Warrants.
Other attributes are inherited from Resource Type (Table 3-1)				

Deleted: The exponent of the Quantity …

Deleted: For example, a

Deleted: denominated

Deleted: QuantityScale

Deleted: 3

Deleted: Warrant Designator

Products with differing Warrants are different Products and therefore traded in different Market Segments.

As non-normative examples, if a Party wishes to buy energy with a *Green Warrant* (however defined) then the Party, not the Market, is responsible for defining its trading strategies if the warranted Product is not available. Similarly, a Party that wishes to buy or sell Neighborhood Solar Power is responsible for submitting Tenders that expire in time to make alternate arrangements, or in time to cancel Tenders before fulfillment. This specification establishes no expectation that the Market engine will address these issues automatically.

Warrants are defined in [EMIX], and CTS permits Warrants to support this complexity if desired, but not described in this specification. A Market MAY define a list of Warrants. Warrants were defined in [EI] as additional non-essential characteristics of a Resource such as how it was produced, or an attribute of regulatory interest. Warrants are defined in the Market but are offered per Segment.

Deleted: and Warrant Designators.…

3.1.3 Defining Instrument

A Market Segment trades Instruments for a single Product. In CTS, an Instrument is a Product delivered for a specific duration beginning at a certain time. CTS includes Duration explicitly in both the Tender and the Quote. The Instrument follows the pattern defined in WS-Calendar—a Resource bound to a Duration (forming a Product) and the Product bound to a Starting DateTime.

The Instrument Start time added to a Product creates an Instrument. See Figure 3-2.

Table 3-3: Specifying the Instrument

Attribute	Type	FIX Field	Meaning	Notes
Instrument Start	Instrument Type	EventType (865), EventDate (866), EventTime (1145)	Starting Date & Time	A start time completes the specification of Product into a tradeable Instrument The Start Time serves a purpose similar to that of the FIX repeating group EvntGrp with EventType(865)=21 (Delivery start time)
The fields are inherited from Product Type, Table 3-2				

Deleted: Instant

Deleted: Not in FIX

Every Tender, Transaction, and Quote is to buy or sell a quantity of an Instrument.

Within a Segment, the Start Date and Time uniquely identifies an Instrument. Because an off-market Segment, sometimes known as an Over The Counter (OTC) Segment can transact products of any Duration, Tenders, Quotes, and Transactions all use the Segment identifier, the Start Time, and the Duration to identify the Instrument and Product.

3.1.4 Summary of Instrument Specification

A UML class diagram for Instrument showing inheritance is in Figure 3-4 below:

Deleted: model

Deleted: the

Deleted: all

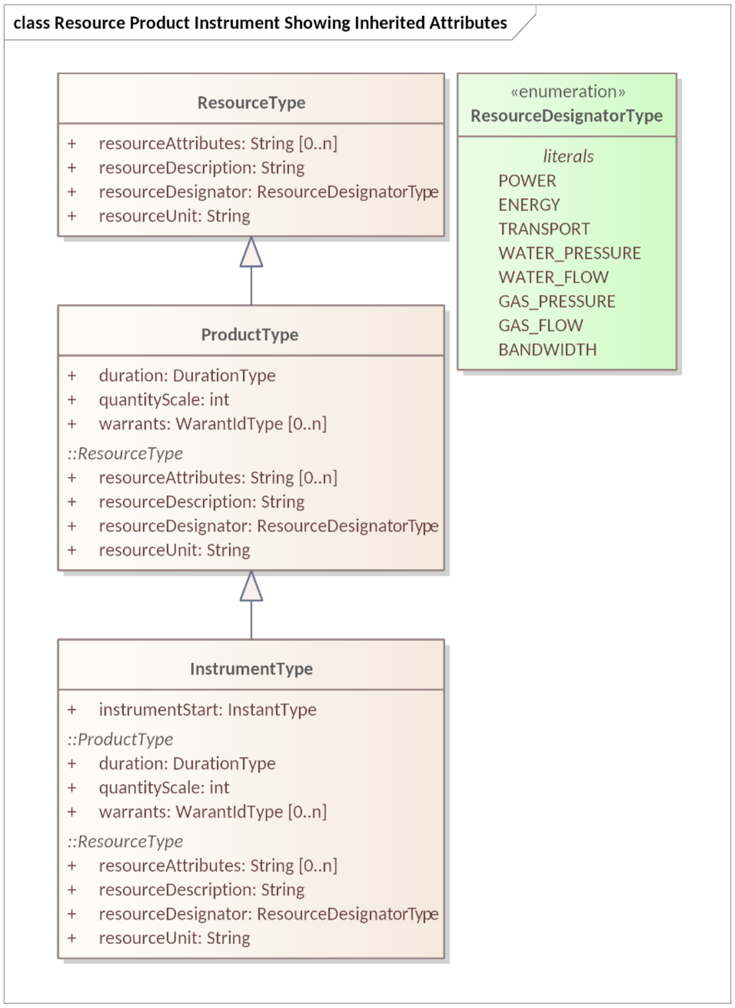
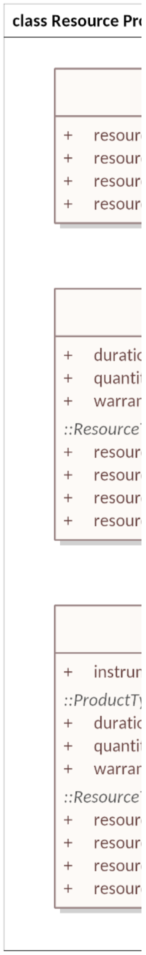

class Resource Product Instrument Showing Inherited Attributes
ResourceType
+ resourceAttributes: String [0..n]
+ resourceDescription: String
+ resourceDesignator: ResourceDesignatorType
+ resourceUnit: String
«enumeration»
ResourceDesignatorType
literals
POWER
ENERGY
TRANSPORT
WATER_PRESSURE
WATER_FLOW
GAS_PRESSURE
GAS_FLOW
BANDWIDTH
ProductType
+ duration: DurationType
+ quantityScale: int
+ warrants: WarantIdType [0..n]
::ResourceType
+ resourceAttributes: String [0..n]
+ resourceDescription: String
+ resourceDesignator: ResourceDesignatorType
+ resourceUnit: String
InstrumentType
+ instrumentStart: InstantType
::ProductType
+ duration: DurationType
+ quantityScale: int
+ warrants: WarantIdType [0..n]
::ResourceType
+ resourceAttributes: String [0..n]
+ resourceDescription: String
+ resourceDesignator: ResourceDesignatorType
+ resourceUnit: String

Figure 3-4 UML Class Diagram for Instrument showing Inheritance from Resource & Product

3.2 CTS Streams: Expressing Time Series

Resource Markets are based on time-of-delivery. It is often useful to convey requests and information about consecutive durations. This specification uses the simplified pattern described in WS-Calendar [Streams], that is, common information followed by a repeating set of information for each consecutive Interval. Each Interval uses a common Duration. All Intervals in a Stream are consecutive.

Deleted:



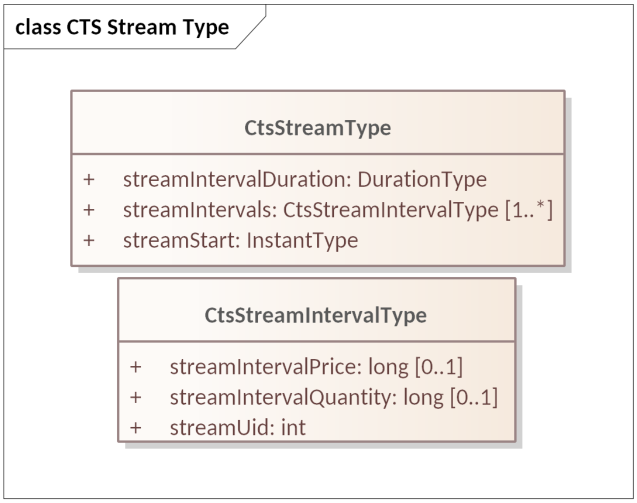

class CTS Stream Type
CtsStreamType
+ streamIntervalDuration: DurationType
+ streamIntervals: CtsStreamIntervalType [1..*]
+ streamStart: InstantType
CtsStreamIntervalType
+ streamIntervalPrice: long [0..1]
+ streamIntervalQuantity: long [0..1]
+ streamUid: int

Figure 3-5: UML Class Model for CtsStream and the Stream Intervals

The response to a request for a stream includes a stream.

Deleted: is

For example, the common information in a TenderStream, derived from the CTS Stream, is the Product and the Start DateTime for the first element of the Stream. The Product specifies Resource and Duration. The consecutive intervals in the CtsStream begin with the Start DateTime for the specified Duration. The second Interval has an implied start of the end of the first Interval. The third Interval has an implied start of the end of the second Interval…and so on.

Each interval carries what can be considered a *local UID*.[10]

Several Facets request a CtsStream in the response. They are:

- Position Facet
- Delivery Facet

Certain payloads may include a CtsStream, including:

- Tender Facet (see *"Interval Tenders and Stream Tenders",* Section 5.3.1)
- Quote and Negotiation Facet *(see Stream Quote)*

Table 3-4: Specifying the Stream

Deleted: shows payloads for the Position and Delivery Facets as an example of the pattern for requesting and responding with streams.¶

Attribute	Type	FIX Field	Meaning	Notes
Stream Interval Duration	Duration Type	*Not in FIX*	The interval Duration for each Stream element.	As defined in **[WS-Calendar]** Optional if inherited from message containing Stream
Stream Start	Instant Type	*Not in FIX*	Starting Date & Time for the first element in the series of Intervals.	After the first Interval, each Interval starts when the preceding Interval finishes
Stream Interval Price Value	Long	Price (44)	Price per Unit during Interval	Optional depending upon purpose of message including Stream

[10] Certain serializations for payloads do not guarantee order, so a small integer serves as a unique identifier for each interval.

Attribute	Type	FIX Field	Meaning	Notes
Stream Interval Quantity Value	Long	OrderQty (38)	The Quantity of the Product during the Interval	Optional depending upon purpose of message including Stream
StreamUID	Integer	*Not in FIX*	Unique identifier for each interval; local to the Stream instance.	Certain deserializations do not guarantee order -- the UID enables reconstructing the order. A simple integer suffices as a sortable UID for streams.

3.3 The Bounding Interval Pattern in CTS

The CTS requests may include a Bounding Interval. The response is typically all Intervals (CTS Stream Intervals, or Instruments) that are contained within the Bounding Interval including those which align with the ends of the Bounding Interval.

More formally, given a request including a Bounding Interval the request will return information on all Instruments or Stream Intervals within the Bounding Interval whose start is at or later than the Bounding Interval start and whose end point is at or before the end of the Bounding Interval.

One common pattern (see e.g. Figure 7-2 and Figure 8-2) is to request information for a Bounding Interval where the response is a CtsStream.

Deleted: See Section “”.¶

The information within each Interval varies per message type. For example, a StreamQuote will put the Price and Quantity in each interval. A Delivery (metering) payload will put only the Quantity in each Interval.

4 Party Registration Facet

A valid Party ID is required to interact with a Market and is included in most payloads.

Party Registration is described in EI. This facet describes the messages necessary for an actor to register and obtain a Party ID to participate in a Market.

EiCreateParty associates an actor with a Party ID and informs the Market of that ID. CTS makes no representation on whether that ID is an immutable characteristic, such as a MAC address, a stable network address, such as an IP, or assigned during registration.

EiRegisterParty names the exchange of information about an actor that enables full participation in a CTS Market. It may exchange information needed for financial transfers including, perhaps, reference to an existing customer or vendor ID, or proof of financial bond for large participants, or issuance of crypto-tokens, or any other local market requirements. A Registered Party is ready to be a full participant in the local Market.

Cancel Party Registration removes a Party from the Market. It may include final settlement, cancellation of outstanding Tenders, backing out of future contracts, or other activities as defined in a particular CTS Market.

Aside from the business services as described, Party Registration may have additional low-level requirements tied to the protocol itself used in a particular implementation based on CTS.

This specification does not attempt to standardize these interactions and messages beyond naming the Register Party facet. A more complete discussion can be found in the [EI] specification.

Some Markets MAY wish to associate one or more measurement points with a Party. Such measurement points could be used to audit Transaction completion, to assess charges for using uncontracted-for-energy, etc. Measurement points are referenced in *Section 8 “The Delivery Facet”*, Markets that require this functionality may want to include an enumeration of Measurement Points in Party Registration.

An implementation is not required to use the Party Registration Facet. For example, if uniqueness and universality are satisfied, any assignment of Party IDs should work.

5 The Tender Facet (Order Messages)

Deleted: messages

A party wishing to buy or sell submits an order ("Tender") using the Tender Facet. The Service descriptions and payloads in [EI] are simplified and updated in CTS. The FIX Protocol classifies Tenders as Orders. Simple Tenders are handled as what the FIX Protocol would describe as Single Leg Orders with related messages as defined in the FIX category SingleGeneralOrderHandling.

Deleted: one of the Order messages.…

Deleted: Simple General

Deleted: and are handled

Deleted: FIX SimpleGeneralOrderHandling…

5.1 Messages for the Tender Facet

Parties exchange Order messages to find or create a Transaction. The Tender Facet payloads are shown in Table 5-1.

Tenders and transactions are artifacts based on **[EMIX]** artifacts, suitably flattened and simplified, and which contain schedules and prices in varying degrees of specificity or concreteness.

Table 5-1: Tender Facet Payloads

Facet	CTS Initial Message	CTS Response Message	Meaning
EiTender	EiCreateTender	EiCreatedTender	A Party sends a Create message containing one or more Tenders to requesting that the [Market][11] create a Tender. The [Market] returns the Created acknowledgement or returns errors, and when successful returns the Market-assigned ID for the submitted Tender
EiTender	EiCancelTender	EiCanceledTender	Cancel one or more Tenders

In the FIX Protocol, an Order is "completed" when it is fully filled, when it is cancelled, or when it expires. FIX also supports the replacement of orders to change some of its attributes. CTS does not permit replacing tenders, instead requiring that a Party cancel a tender and submit a new one. If a Tender is already partially filled, cancellation cancels only the unfilled portion.[12]

Deleted: specification, a Tender…

Deleted: is replaced

5.1.1 Illustrative Narrative on Tenders [Non-Normative]

For example, Party A submits a Tender 1 to buy 100 kWh over an hour. A Tender from Party B for 45 kWh matches Party A's Tender and the Market creates a Transaction (see Section **6**, *"The Transaction Facet"* for a discussion of Transactions). A Tender from Party C for 35 kWh matches Party A's Tender and the Market creates a Transaction. Party A's Tender 1 remains on the market with 20 kWh remaining. If Party A wishes to increase the price offered to get the 20 kWh for a critical operation, Party A must cancel Tender 1, with 20 kWh remaining, and submit a Tender 2 offering a new price. Cancelling Tender 1 does not invalidate either of the two completed Transactions.

5.2 Interaction Patterns for the Tender Facet

Figure 5-1 presents the UML sequence diagram for the EiTender Facet. Note that while [EI] defines a message EIDistributeTender, CTS uses the Negotiation Facet (Section 9, *"The Negotiation Facet"*) and Ticker Subscriptions (Section 11, *"Tickers"*) to accomplish similar purposes.

[11] See Section 9 *"The Negotiation Facet"* and Section 13.1, *"Market Mechanisms"* for discussions where the message target may not be the Market.

[12] This avoids a potential race condition in variable latency distributed systems.

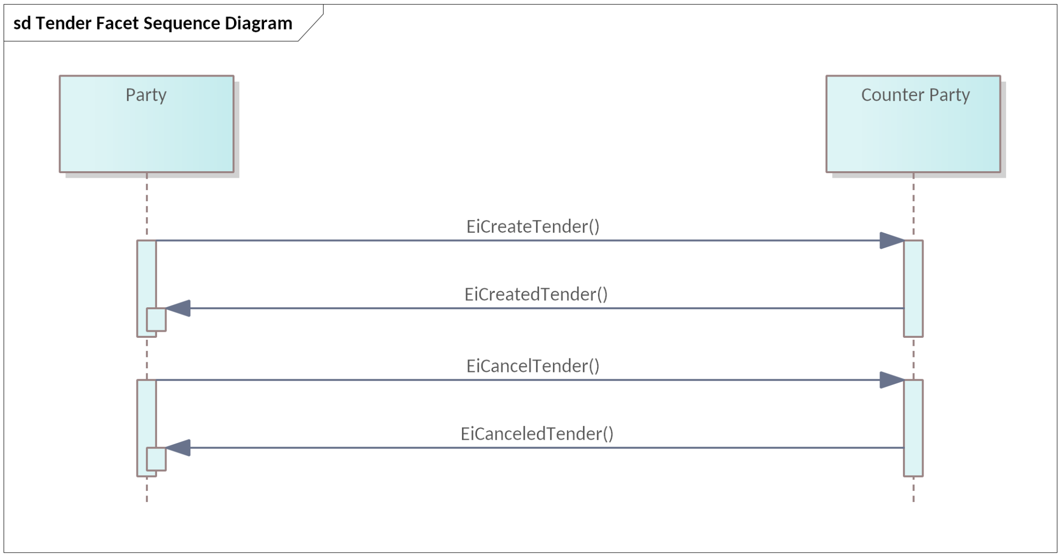


Figure 5-1: UML Sequence Diagram for the Tender Facet

5.3 Information Model for the Tender Facet

The information model for the Tender Facet artifacts follows that of [EMIX] but flattened and with Product definition implied by the implementation. See Section 5.6 Message Payloads for the Tender Facet below.

The Tender and Quote and RFQ classes share most attributes in common. Accordingly, a superclass *Tender Base* holds those common attributes as shown in Figure 5-2.

TenderBase is an abstract class, so no object can be of that class.

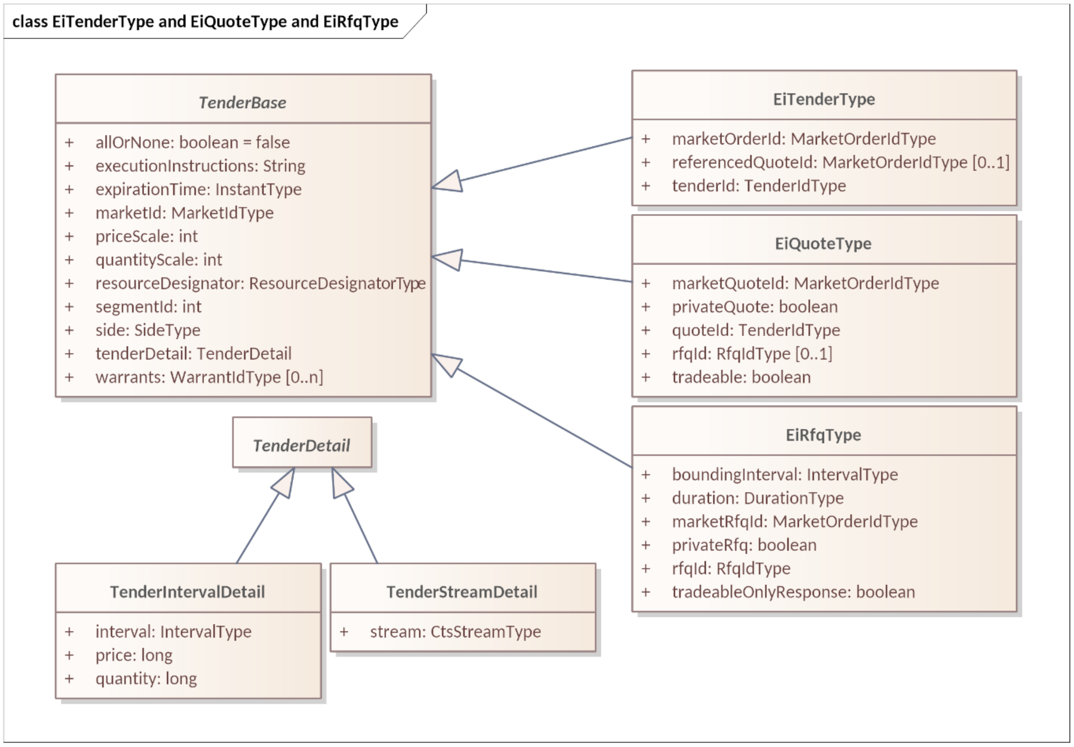
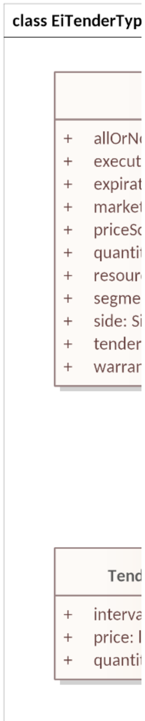

class EiTenderType and EiQuoteType and EiRfqType
TenderBase
+ allOrNone: boolean = false
+ executionInstructions: String
+ expirationTime: InstantType
+ marketId: MarketIdType
+ priceScale: int
+ quantityScale: int
+ resourceDesignator: ResourceDesignatorType
+ segmentId: int
+ side: SideType
+ tenderDetail: TenderDetail
+ warrants: WarrantIdType [0..n]
EiTenderType
+ marketOrderId: MarketOrderIdType
+ referencedQuoteId: MarketOrderIdType [0..1]
+ tenderId: TenderIdType
EiQuoteType
+ marketQuoteId: MarketOrderIdType
+ privateQuote: boolean
+ quoteId: TenderIdType
+ rfqId: RfqIdType [0..1]
+ tradeable: boolean
EiRfqType
+ boundingInterval: IntervalType
+ duration: DurationType
+ marketRfqId: MarketOrderIdType
+ privateRfq: boolean
+ rfqId: RfqIdType
+ tradeableOnlyResponse: boolean
TenderDetail
TenderIntervalDetail
+ interval: IntervalType
+ price: long
+ quantity: long
TenderStreamDetail
+ stream: CtsStreamType

Figure 5-2 UML Class Diagram Showing Commonality between Tender, Quote, and RFQ

Figure 5-3 shows all attributes for EiTenderType and their sources.

Deleted:



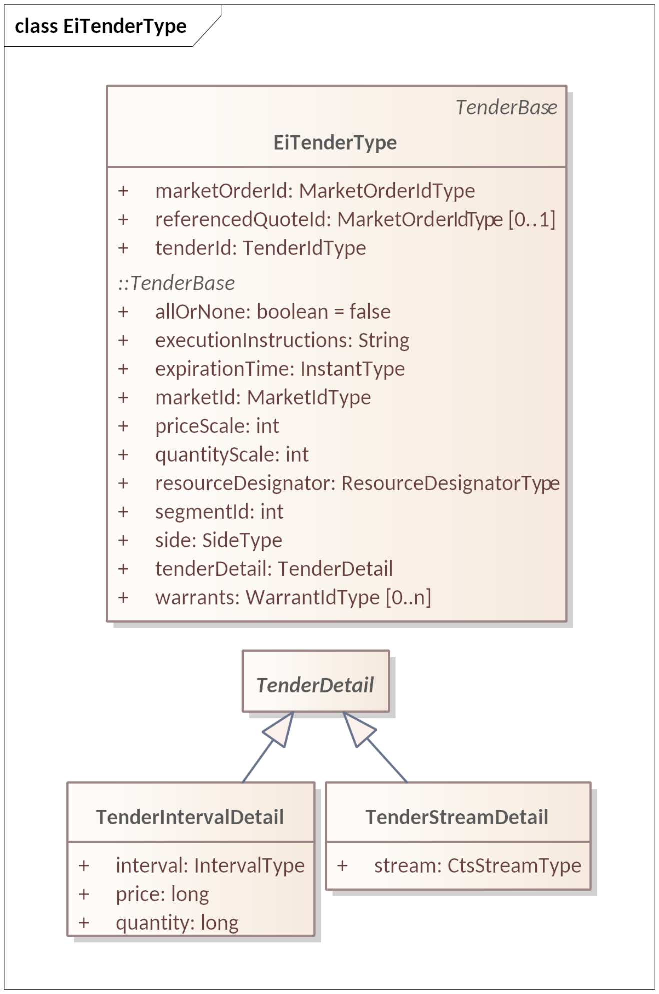


Figure 5-3 UML Class Diagram showing EiTenderType

EiTenderType inherits from TenderBase, which holds the common attributes between Tender, Quote, and RFQ.

Attributes used in Tenders and TenderBase are shown in Table 5-2 and Table 5-3.

Of the attributes in Table 5-2 Tender ID and Referenced Quote ID (Referenced Quote Id) are unique to EiTenderType; the others are inherited from Tender Base and shared with EiQuoteType and EiRfqType. See Section 9, “*The Negotiation Facet”*, for a discussion of Quotes and Requests For Quotes.

Table 5-2: EiTender Attributes

Attribute	Type	FIX Field	Meaning	Notes
Market Order ID	Market Order ID Type	OrderID(37)	A market-assigned unique identifier for an Order (Tender in CTS)	

Deleted: OrderId

Attribute	Type	FIX Field	Meaning	Notes
Referenced Quote ID	UID	QuoteMsgID (1166)	ID of the Tradeable Quote to which this is a response.	Optional. If Quote ID is not known to the Market Segment, or if the referenced Quote has expired, then the Tender is rejected.
Tender ID	Tender ID Type	ClOrdId(11)	An ID for this Tender generated by the submitting Party	
Other attributes are inherited from TenderBase—See Table 5-3				

The complete description of the Interval for a Tender is in the TenderDetail—either an Interval with a price and quantity, or a CtsStream with that information for each Stream Interval.

Deleted: Cts Stream

While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off-market interactions.

Table 5-3 Tender Base Attributes

Attribute	Type	FIX Field	Meaning	Notes
All or None	Boolean	In FIX, this is one among many Execution Instructions	All or none of the tendered or quoted amount must be traded.	In Energy Interoperation 1.0 this was called *IntegralOnly*. In CTS, this is promoted from Execution Instruction to top-level attribute.
Execution Instructions	String	ExecInst (18)	FIX Supports many instructions for how to execute an order.	See Table 5-4 below. Modeled as a String in CTS.
Expiration Time	Instant Type	ExpireTime (126)	The Tender or Quote expires at the specific time.	Always expressed in UTC
Market ID	Market ID Type	MarketID (1301)	Identifies the Market	Note that in FIX, this is generally a formal identifier (e.g., “NYSE”). If the market is a house, there is no place to look this up. There is always a UID for a Market.
Price Scale	Integer	Not in FIX	A multiplier for the Price	Note that Price Scale is specific to this Tender Base, and not necessarily to Markets or Segments in which the Tender Base. may be used.
Quantity Scale	Integer	UnitOfMeasure (996) UnitOfMeasure Quantity (1147)	A scale factor on the Resource unit for this Market	See Table 3-2: Defining the Product. Note that the Tender Quantity Scale is specific to this Tender Base, and not necessarliy to Markets or Segments in which the Tender Base may be used.

Deleted: a tender

Deleted: Defined

Deleted: Must match the price Scale of the Segment.

Deleted: For example, “mega” vs “kilo” vs “femto-”

Attribute	Type	FIX Field	Meaning	Notes
Resource Designator	Resource Designator	Not in FIX	Identifier of the Resource being offered (Optional in many markets)	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions.
Segment ID	Integer	MarketSegment ID(1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Side	Side Type	Side(54)	Whether the Tender is to buy or to sell the Product	Buy or Sell side
Tender Detail	Tender Detail	Not in FIX	Unit price and quantity for this tender	May be Interval or Stream as permitted
Tender ID	UID	ClOrdId(11)	ID as submitted to Market	Identifies Tender until Market Order ID is assigned by Market
Tender Interval Detail	Tender Interval Detail	Not in FIX	Interval, price and quantity for this tender	Used in Interval Tender
Tender Stream Detail	Tender Stream Detail	Not in FIX	Stream of consecutive Intervals with Prices and Quantities	Sometime referred to as a Load Curve in Power Markets.
Warrants	Warrant ID Type	Not in FIX	Reference to Warrants as defined in the Market	If used, see Warrants in Tenders, Section 5.3.3.
The following attributes are in Tender Interval Detail or Tender Stream Detail—See Figure 5-3				
Interval	Interval Type	Not in FIX	Start Instant for Product delivery together with Duration of delivery. Part of Instrument	While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off-market interactions.
Price	Long	Price(44)	The unit price for the Product being Tendered	Amount is the product of Price and Quantity. Note that Price is subject to the Price Scale for this Tender Base.
Quantity	Long	OrderQty(38)	The quantity of the Product being Tendered	Note that Quantity is subject to the Quantity Scale for this Tender Base. Quantity must meet the Quantity Scale and Round Lot requirements of the Segment in which the containing Tender, Quote, or RFQ is to be traded . (see Table 13-5)

Deleted: defined

Moved (insertion) [1]

Deleted: UID

Deleted: defined

Deleted: defined

Deleted: defined

Deleted: Warrant

Deleted: defined

Moved up [1]: Integer

Deleted: defined

Deleted: , and Quantity to the Quantity Scale.

Deleted: Must

Deleted: .

Attribute	Type	FIX Field	Meaning	Notes
Stream	CTS Stream Type	Not in FIX		Attribute of TenderStreamDetail—see Figure 5-3.

Deleted: defined

5.3.1 Interval Tenders and Stream Tenders

The most common Tender is the simple Interval Tender, that is, an offer for a Product in a single interval beginning at a specific date and time.

In financial markets, a *multi-leg order* is submitted for securities that are made up of multiple securities, known as legs. The legs are not traded individually. This specification describes a specialized type of multi-leg order for use in in some Market Segments which we term a Stream Tender. A Stream Tender defines a consecutive series of Intervals of identical Duration. The price and quantity tendered must be specified for each Interval.

For example, an industrial customer in a power market may intend to buy power to support a long running process. In power markets, such a sequence of power use is sometimes referred to as a *load curve*.

Such multi-leg orders are expressed using a CtsStream (see 0, “

CTS Streams: Expressing Time Series”). While the information contained in a Stream Tender can be mapped precisely to a group of Interval Tenders, multi-leg semantics and processing of the related tenders leads to a Stream Tender.

Not all Market Segments permit Stream Tenders; some may require them. A Party submits a Stream Tender, when permitted or required, just as a Party submits an Interval Tender. A Market responds to the submission of a Stream Tender, when permitted or required, just as it responds to an Interval Tender.

Partys may submit Stream Tenders only to Market Segments that specifically permit or require them; submission to all other Segments are forbidden. See Section 13, “Market Structure Reference Data: Market, Segment, and Session Subscriptions”.

Market Segments that support Stream Tenders SHALL also support Stream Quotes (if they support Quotes) and Stream Transactions. See Section 9, “The Negotiation Facet”, for a discussion of Quotes.

5.3.2 Execution Instructions

FIX supports many Execution Instructions, while CTS restricts them to a reduced set.

Future versions of CTS may incorporate additional Execution Instructions into future versions of CTS.[13].

For example, the the following instructions could all be on the same order:

- Cross is forbidden.
- Reinstate on system failure.
- Cancel on trading halt.

Deleted: These are modeled as a string using single letters for each FIX Execution Instruction Code separated by a space.

Deleted: string *H K A* indicates …

Table 5-4 presents a subset of the FIX Execution Instructions permitted for use in version 1.0 of CTS.

[13] Segment Reference Data includes which Execution Instructions are supported.

Table 5-4: Trading Instructions

Instruction	FIX Code		Notes
No cross	A		Tender is cancelled after any market transition (See 13.4, “Trading Session Data”)
OK to cross	B		Cross is Permitted. (See 13.4, “Trading Session Data”)
All or none – AON	G		Ignored in deference to the AllOrNone attribute.
Reinstate on system failure	H		Mutually exclusive with Q and l (lower case L).
Reinstate on trading halt	J		Mutually exclusive with K and m.
Cancel on trading halt	K		Mutually exclusive with J and m.
Cancel on system failure	Q		Mutually exclusive with H and l (lower case L).
Cancel if not best	Z		Cancel if order is not immediately matchable
Ignore price validity checks	c		
Suspend on system failure	l		Mutually exclusive with H and Q.
Suspend on trading halt	m		Mutually exclusive with J and K.

Deleted: Abbreviation

Deleted Cells

Deleted: [NoCross]

Deleted: [OKToCross]

Deleted: [AllOrNone]

Deleted: [ReinstateOnSystemFailure]

Deleted: [ReinstateOnTradingHalt]

Deleted: [CancelOnTradingHalt]

Deleted: [CancelOnSystemFailure]

Deleted: [CancelIfNotBest]

Deleted: [IgnorePriceValidityChecks]

Deleted Cells

Deleted: [SuspendOnSystemFailure]

Deleted Cells

Deleted: [SuspendOnTradingHalt]

5.3.3 Use of Warrants in Tenders

Warrants increase the specificity of Product (and Instrument). A Buyer who does not specify a Warrant will be satisfied by Delivery of a Product whether or not it has a Warrant. A Buyer who requests Product with a Warrant will only be satisfied by Delivery of a Product that has that Warrant.

Consider a buyer who wishes to buy a package of coffee beans and a buyer who wishes to buy a package of organic coffee beans. The word “Organic” on the label serves as a Warrant. The first buyer will buy solely on price, and is indifferent to seeing the word “Organic” on the label. The second buyer will choose only from among those packages with the warrant “Organic” on the label.

When a Tender on the Buy side specifies a Warrant, it must be rejected by any Market Segment that does not include that Warrant. A Tender on the Sell side that specifies a Warrant may be accepted by any Segment where the same Resource and Duration are traded. Conversely, a Tender on the Sell side without a Warrant must be rejected by any Segment that specifies a Warrant.

5.4 Contingent Tenders

FIX permits multiple Orders submitted in a single message. The FIX List Order bundles multiple Orders with a common instruction that influences how fulfilling each Order affects the other Orders. A CTS Market Segment either forbids or requires the use of Contingent Tenders. Tender Contingency Types in CTS are based on the values of the FIX field ContingencyType(1385).

Deleted: Tenders

Deleted: Tenders (

Deleted:)

Deleted: Tender

Deleted: Tenders

Deleted: defined by

5.4.1 Illustrative Narrative on Contingent Tenders [Non-Normative]

The Contingency Type describes how the other Tenders in the List are affected by the acceptance of any one Tender in the Market. A Party submitting a List with atMostOne = True is willing to accept whatever Tender matches the Transaction that returns from the Market. In CTS Version 1, the FIX-defined Contingency OCO or “One Cancels the Other” is expressed as a Boolean *atMostOne*.

Stream Tenders are a special case. Stream Tenders (Load Curves) support business needs such as acquiring power for a long-running industrial process. The sub-Tenders that compose a Stream Tender are always treated as “All or None”.

A Party MAY wish to probe the market to make a more nuanced decision. This may include choosing one of several options. A decision to schedule a long-running process may depend upon being able to acquire a specific load curve over the entire schedule. A party that requires such complex contingent behavior should use the Negotiation Facet (section 9) to obtain Tradeable Quotes, and then make its own choices based on those Quotes.

5.5 Rejecting a Tender and Tender Responses

A Market may reject a Tender that violates market rules or which, if transacted, would violate the market’s integrity and other constraints (e.g. liquidity goals). Rejection Reasons include but are not limited to:

- Tender exceeds price limits on the potential transaction.
- Tender exceeds total value limits on the potential transaction.
- Tender violates total quantity limits for this Market Segment.
- Party is not in good standing with the Market.
- Tender violates lot size requirements of the Market Segment.
- Tender violates starting time requirements for instruments in the Market Segment.
- Market Segment is not open.
- Instrument is prior to temporal trading limits for this Market Segment.
- Instrument is past temporal trading limits for this Market Segment.
- Tender is incomplete or corrupt.
- Referenced Quote not found.
- Referenced Quote has expired.

Details for rejection MAY be included in the EiResponse included in the EiCreateTenderPayload; the details apply to other Create payloads as well. See Section 2.4 Responses.

Deleted: EiCreatedTender Payload.…

The optional Response Description string is for implementation-defiined additional information..The EiResponse is used in a similar way for other response payloads.

5.6 Message Payloads for the Tender Facet

Figure 5-4 is a **[UML]** class diagram for the payloads for the Tender Facet operations. Note that each operation supports a Tender Set, and any set may consist of any number of Tenders, Interval or Stream.

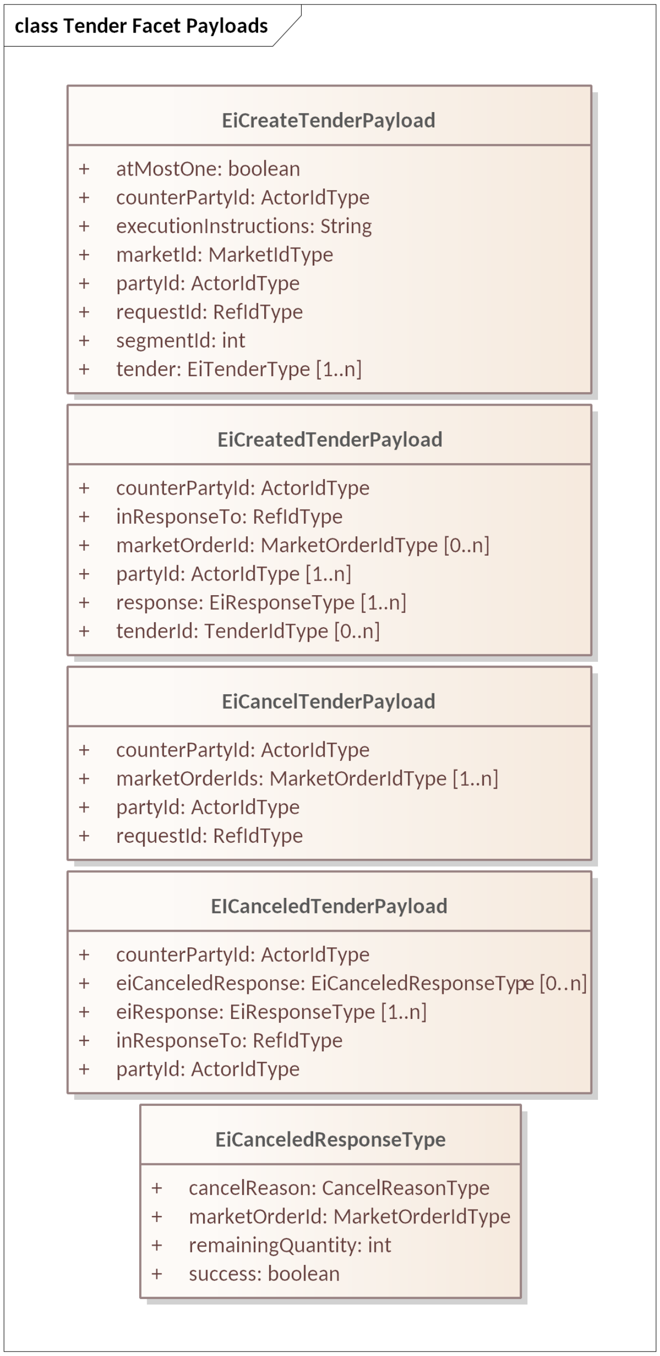

class Tender Facet Payloads
EiCreateTenderPayload
+ atMostOne: boolean
+ counterPartyId: ActorIdType
+ executionInstructions: String
+ marketId: MarketIdType
+ partyId: ActorIdType
+ requestId: RefIdType
+ segmentId: int
+ tender: EiTenderType [1..n]
EiCreatedTenderPayload
+ counterPartyId: ActorIdType
+ inResponseTo: RefIdType
+ marketOrderId: MarketOrderIdType [0..n]
+ partyId: ActorIdType [1..n]
+ response: EiResponseType [1..n]
+ tenderId: TenderIdType [0..n]
EiCancelTenderPayload
+ counterPartyId: ActorIdType
+ marketOrderIds: MarketOrderIdType [1..n]
+ partyId: ActorIdType
+ requestId: RefIdType
EICanceledTenderPayload
+ counterPartyId: ActorIdType
+ eiCanceledResponse: EiCanceledResponseType [0..n]
+ eiResponse: EiResponseType [1..n]
+ inResponseTo: RefIdType
+ partyId: ActorIdType
EiCanceledResponseType
+ cancelReason: CancelReasonType
+ marketOrderId: MarketOrderIdType
+ remainingQuantity: int
+ success: boolean

Figure 5-4 UML Class Diagram for Tender Facet Payloads

The Market Order ID is assigned by the Market on receipt of a Tender. The Market makes no assumption that the Tender ID (FIX field ClOrderID(11)) submitted as part of the Tender is unique across all Parties in the Market. The Market responds with a Market Order ID for each Tender ID submitted. The submitting

Deleted: Client Order

Party should record this Market Order ID, as it will be used in any Transactions awarded by the Market, and is required to cancel any Tender.

Specific Market Segments may limit all Tender submissions to either Interval Tenders or to Stream Tenders or may accept both. Specific Market Segments may restrict each Tender Set to all Interval Tenders or all Stream Tenders. Specific Market Segments may limit the cardinality of a Tender Set to any count. In the absence of such Segment specification, to support minimal interoperability, Interval Tenders are permitted, Stream Tenders, and the cardinality of each Tender Set is limited to one.

See Section 13 for details.

The following tables describe the attributes for the Tender Facet Payloads.

Table 5-5 EiCreateTenderPayload Attributes

Attribute	Type	FIX Field	Meaning	Notes
At Most One	Boolean	Contingency Type (1385)	Used to express alternatives, only one of which is to be effective	See Trading Instructions in Table 5-4. First match cancels other Tenders.
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the Counterparty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Execution Instructions	String	ExecInst (18)	Execution Instruction.	Execution instructions apply to each Tender in the payload. Multi-leg (Stream) Tenders are always All-or-None.
Market ID	Market ID Type	MarketID (1301)	Identifies the Market	Note that in FIX, this is generally a formal identifier (e.g. "NYSE"). If the market is a house, there is no place to look this up. There is always a UID for a Market.
Segment ID	Integer	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender is made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
Request ID	Ref ID	ClOrderID (11)	An identifier for this Create Tender Payload	The FIX Protocol makes no assumption that IDs submitted by market participants will actually be complete.
Tender	Ei Tender Type		Tenders requested to be created	One or more Tenders per Table 5-2: EiTender Attributes.

Deleted: See

Table 5-6 EICreatedTenderPayload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
In Response To	Ref ID	ClOrderID (11)	An identifier for Create Tender Payload to which this is a response	
Market Order ID	UID	OrderID (37)	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages.

Attribute	Type	FIX Field	Meaning	Notes
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender is made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
Response	EiResponse Type	OrdRejReason(103)	Specific error responses	See Section 2.4
Tender ID	Tender ID Type	ClOrderID (37)	The Tender ID that was used to submit the Tender to which this is a response	While UIDs should be truly unique, with a mix of technologies and possible faulty implementations in low-end devices, CTS follows the FIX Protocol in assuming that Customer Order is only unique for this Customer today.

Table 5-7 EiCancelTender Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Party ID	Actor ID	PartyID (448)	Actor ID for the Party that created the Tender	
Request ID	Ref ID	ClOrderID (11)	An identifier for this Cancel Tender Payload	
Market Order ID	UID	OrderID (37)	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages. As defined in Section 5.

Deleted: Not in FIX

Table 5-8 EiCanceledTenderPayload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender was made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
In Response To	Ref ID	ClOrderID (11)	An identifier for the Cancel Tender Payload to which this is a response	

Moved down [2]: Not in FIX

Attribute	Type	FIX Field	Meaning	Notes
EiCanceled Response	EiCanceled Response Type	Not in FIX	Detailed response for each tender included in the EiCancelTender Payload	See Section 5.5.
EiResponse	EiResponse Type	CxlRejReason(102)	Specific error responses	See Section 2.4

Deleted: Ei Canceled

Deleted: Canceled

Deleted: Not in FIX

6 The Transaction Facet (Execution)

This section presents the Transaction Facet, used by the Market to notify of the creation of Transactions. FIX terms the matching of a Buyer and a Seller as a “Trade” or “Execution”. CTS follows EI (and the term transactive energy) in naming it a Transaction.

In the general case, the Market notifies each Party of the creation of a Transaction when two Tenders match as discovered by the Market’s internal matching engine. To protect participant privacy, the market MAY use the MarketID as the counterparty to each Party receiving the Notification.

Unlike in financial markets, the market operator must cooperate with relevant system operators to enforce flow limits imposed by physical infrastructure limits. For example, a substation or distribution cable will have physical limits for power transferred during a given Interval. The reasons and mechanisms for such an enforcement are out of scope for CTS.

See Section 9, *“The Negotiation Facet”* for a discussion of Transactions based upon a Tradeable Quote.

All Transactions are committed, that is, they cannot be cancelled or modified under normal market operations. Transactions in aggregate make up the Position. (See Section 7, *“The Position Facet”* for a discussion of Position.) A Party may thereafter choose to sell any or all of its Position in any instrument. Moreover, market and/or segment rules (out of scope for CTS) may limit attempts to sell more than the Party’s current position; in FIX such check may include a pre-trade risk limit or credit limit check.

6.1 Messages for the Transaction Facet

A Transaction is created by a Market or Segment (See Section 13) based on some Mechanism internal to the Market.[14] (See Section 13.1 for what a Party can know of the Mechanism.) When a Market recognizes a potential Transaction, it creates a Transaction ID, and notifies the participating Parties.

Table 6-1: Transaction Facet

Facet	CTS Initial Message	CTS Response Message	Meaning
Transaction	EiCreateTransaction	EiCreatedTransaction	Create and acknowledge creation of a Transaction; typically initiated by the matching engine of the Market Segment.

6.2 Interaction Pattern for the Transaction Facet

Figure 6-1 shows the UML sequence diagram for the EiTransaction Facet.

[14] Some aspects of the market’s mechanism(s) are visible to actors who are trading, generally where the mechanism affects rational bidding strategies. For example, bidding very low in an auction market is reasonable (as you get the clearing price), but bidding very low in an order book market is not (as you may get something like what you offered). See Section 13 and Market Mechanisms.

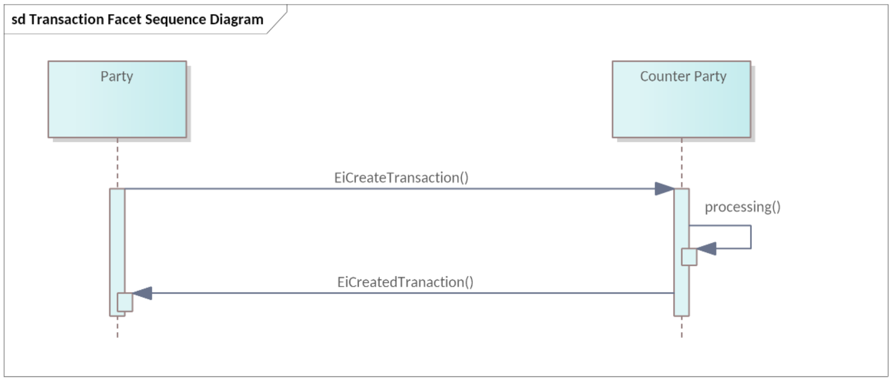


Figure 6-1: UML Sequence Diagram for the EiTransaction Facet

Most Transactions are mediated by a market. The Market matches Tenders, creates a Transaction, and notifies the submitting Parties.

In Off-Market and quote-based Segments (See Section 13), the Parties match Quote and Tender, and inform the Market to create the Transaction. Even in Off-Market and quotation-based Segments, the market operator must still enforce physical or other limitations. Interaction patterns for such Segments are defined in Section 9, *“The Negotiation Facet”*.

6.3 Information Model for the Transaction Facet

The EiTransaction object includes the information in the original EiTender, possibly updated to reflect the actual price and quantity rather than the requested price and quantity.

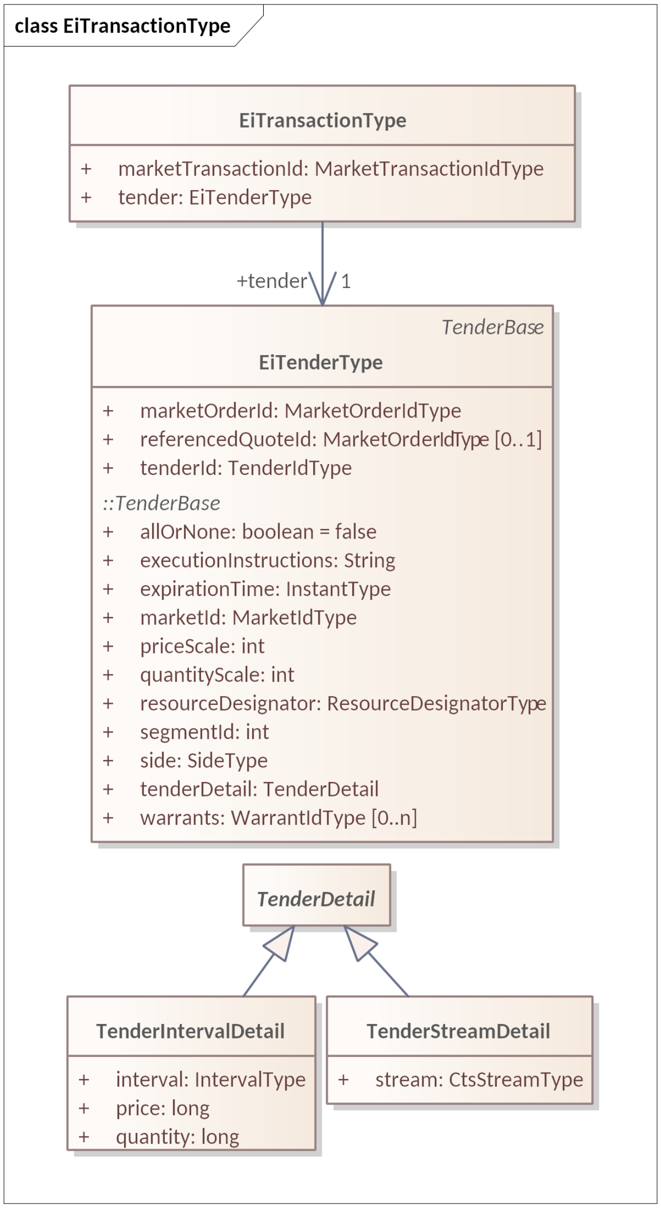


Figure 6-2: UML Class Diagram of EiTransactionType

The attributes of EiTransactionType are shown in Table 6-2.

Table 6-2: EiTransaction Attributes

Attribute	Type	FIX Field	Meaning	Notes
Market Transaction ID	Market Transaction ID Type	TradeID (1003)	ID Assigned this Transaction (Trade) by the Market (Segment)	This is assigned by the actor that performed the match, typically a market segment.
All other attributes are as defined in the Tenderbase, see Figure 5-2				

6.4 Payloads for the Transaction Facet

The **[UML]** class diagram in Figure 6-3 describes the payloads for the EiTransaction facet operations.

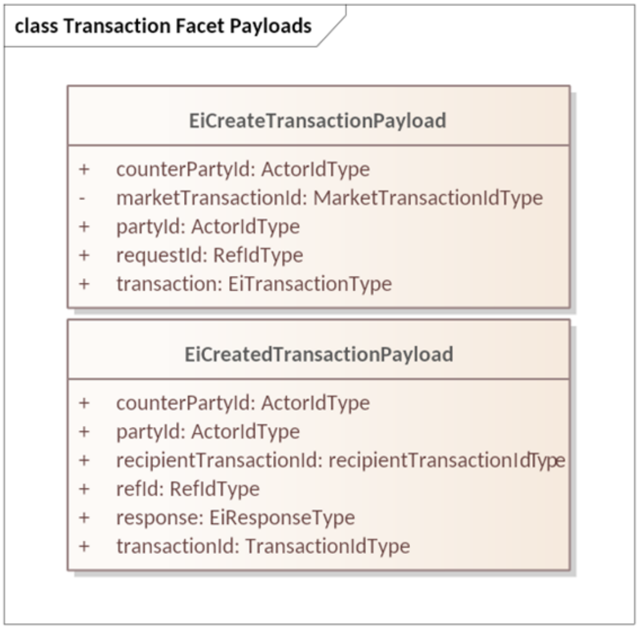

class Transaction Facet Payloads
EiCreateTransactionPayload
+ counterPartyId: ActorIdType
- marketTransactionId: MarketTransactionIdType
+ partyId: ActorIdType
+ requestId: RefIdType
+ transaction: EiTransactionType
EiCreatedTransactionPayload
+ counterPartyId: ActorIdType
+ partyId: ActorIdType
+ recipientTransactionId: recipientTransactionIdType
+ refId: RefIdType
+ response: EiResponseType
+ transactionId: TransactionIdType

Figure 6-3: UML Class Diagram of EiTransaction Facet Payloads

The following tables list the attributes of the Transaction Facet Payloads.

Transactions are produced by a market or actor that performs matches; the resulting Transaction information is sent to the Parties whose Tender(s) are matched. Note that there is not a one-to-one relationship of Tender to Tender, or Tender to Contract. A Tender to buy one hundred might match multiple Tenders to sell ten; this results in multiple Transactions for one Tender. Each Transaction is created by an interaction between a Tender to buy and a Tender to Sell. The Transaction payloads “echo” each Tender to the Party that submitted it to become part of the Transaction.

The Tender included as part of a Transaction payload indicates a buy side or a sell side. When the Transaction indicates “buy”, then the PartyID is that of the Buyer. When the Transaction indicates “sell”, then the PartyID is that of the Seller. The CounterpartyID is the other participant in the Transaction.

Financial markets often designate a “clearing” or “central” counterparty. Privacy concerns, particularly for transactions involving homes, are one reason for using the PartyID of the central counterparty. Some rules may require revealing the identity of certain Parties. For example, the PartyID of a dominant participant such as a distribution serving operator MAY be deemed public information; transactions involving such a designated participant would use the participant’s PartyID in the payload.

When use of a PartyID for the clearing counterparty is required, CTS uses the PartyID of the Market.

Table 6-3 EiCreateTransactionPayload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	PartyID of the Party on the other “side” from the Tender in the payload.	May be the PartyID of the clearing counterparty.
Market Transaction ID	UID	TradeID (1003)	ID assigned by the Market when generating a Trade	Assigned by the Market
Party ID	Actor ID	PartyID (448)	Party ID of the Party on the same “side” of the Tender in the Payload.	Side of the included transaction determines the Party.
Reference ID	String	ExecID (17)	An identifier for this message	
Tender	TenderBase		Price and Quantity for Interval[s] in Transaction	

Deleted: ExecId

Table 6-4 EiCreatedTransactionPayload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	PartyID of the Party on the other “side” from the Tender in the payload.	May be the PartyID of the clearing counterparty.
Market Transaction ID	UID	TradeID (1003)	ID assigned by the Market when generating a Trade	Assigned by the Market
Party ID	Actor ID	PartyID (448)	Party ID of the Party on the same “side” of the Tender in the Payload.	Side of the included transaction determines the Party.
Recipient Transaction ID	Recipient Transaction ID Type	XID	The ID assigned to the received Transaction by the recipient of the associated EiCreateTransaction	
Reference ID	String	ExecID (17)	The Ref ID for the message payload indicating the cleared Transaction	
Response	EiResponse Type		Specific error responses	See Section 2.4

Deleted: ExecId

6.5 Comparison of Tender and Transaction Payloads

In this section we show the payloads for the Tender and Transactive Facets

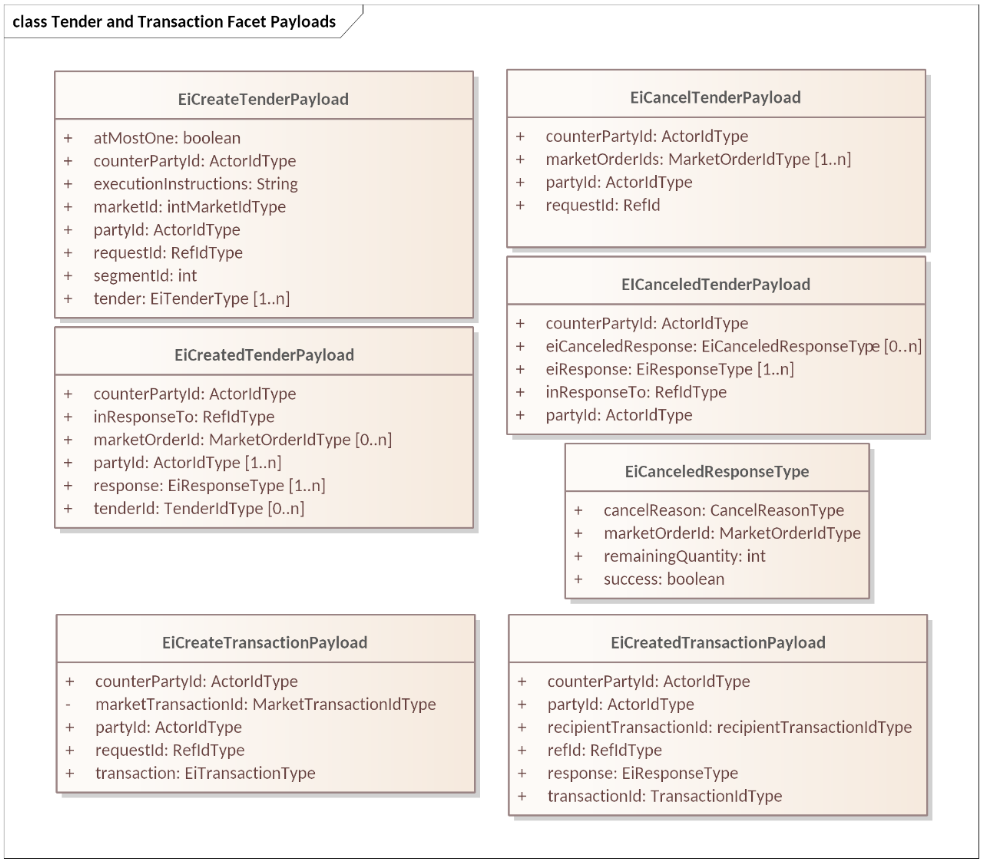

class Tender and Transaction Facet Payloads
EiCreateTenderPayload
+ atMostOne: boolean
+ counterPartyId: ActorIdType
+ executionInstructions: String
+ marketId: intMarketIdType
+ partyId: ActorIdType
+ requestId: RefIdType
+ segmentId: int
+ tender: EiTenderType [1..n]
EiCancelTenderPayload
+ counterPartyId: ActorIdType
+ marketOrderIds: MarketOrderIdType [1..n]
+ partyId: ActorIdType
+ requestId: RefId
EiCreatedTenderPayload
+ counterPartyId: ActorIdType
+ inResponseTo: RefIdType
+ marketOrderId: MarketOrderIdType [0..n]
+ partyId: ActorIdType [1..n]
+ response: EiResponseType [1..n]
+ tenderId: TenderIdType [0..n]
EICanceledTenderPayload
+ counterPartyId: ActorIdType
+ eiCanceledResponse: EiCanceledResponseType [0..n]
+ eiResponse: EiResponseType [1..n]
+ inResponseTo: RefIdType
+ partyId: ActorIdType
EiCanceledResponseType
+ cancelReason: CancelReasonType
+ marketOrderId: MarketOrderIdType
+ remainingQuantity: int
+ success: boolean
EiCreateTransactionPayload
+ counterPartyId: ActorIdType
- marketTransactionId: MarketTransactionIdType
+ partyId: ActorIdType
+ requestId: RefIdType
+ transaction: EiTransactionType
EiCreatedTransactionPayload
+ counterPartyId: ActorIdType
+ partyId: ActorIdType
+ recipientTransactionId: recipientTransactionIdType
+ refId: RefIdType
+ response: EiResponseType
+ transactionId: TransactionIdType

Figure 6-4: UML Diagram comparing Tender and Transaction Facet Payloads

6.6 Off-Market Transactions

While most transactions originate as Tenders submitted to the Market, which some mechanism inside the Market matches, and result in a Transaction created by the Market, there are use cases for bilateral actions that generate a Transaction that did not come through the market.

For example, two parties within a market may choose to transact directly. A party may opt to buy directly from his neighbor's solar power. Another market may permit charity, that is, a donation to the Position of a neighbor. In either case, the Market must register the Transaction so that is can maintain each Party's Position, and so that the Buyer does not get double billed. These transactions may also be referred to as over-the-counter (OTC) agreements.

Off-Market agreements require both parties to report to the Market. The originating Party sends a Tradeable Quote to the Market, including the ID of the counterparty. The simplest means is for one Party to send a targeted Quote (see Section 9, *"The Negotiation Facet",* below) naming the Counterparty in the Quote. The Counterparty then accepts the Quote by submitting a message referencing the Quote Id and including a Tender matching the Tender in the Quote.

Deleted: publish

Some Markets will have specific Market Segments for Off-Market Transactions with specific message patterns. An OTC Market is notable for permitting violations of the Lot Size constraint and of the start time and duration constraints of other market segments. For example, in a Market with a Market Segment with a product of Lot Size 20 kWh and a Duration of one hour, an Off-Market execution could register a transaction of 23 kWh delivered over 27 minutes beginning at 2:48.

See Sections 13.1“Market Mechanisms”.

7 The Position Facet

The Position Facet provides the sum of a Resource transacted for by a Party, positive and negative, for each interval within a possibly larger bounding Interval. For example, a Position may sum up all transactions over the course of a [day]. It is typically requested by an auditor or settlement agent (See Section 8 The Delivery Facet) or by a Party to get information about its own position.

A Party may buy and sell from several Market Segments, perhaps with different Durations. A Party may also transact with specific counterparties in an Over-The-Counter (OTC) market. All of these are part of the Party's position.

In most Resource markets, a Party may also take delivery (see Section 8, The Delivery Facet) which is measured by a meter. But what is the Quantity for this "self-executed" Transaction? This amount is calculated by the difference between Position and Delivery and thereby creates Transactions for the used-but-never-bought Resource; the frequency and nature of such actions are out of scope for CTS.

Deleted: .

There may be other reasons to track Position. A market rule may require a Party designated as a Market Maker to maintain a Position of a certain quantity. A Party representing a Storage System may have specific rules for Position before a weather event. This specification does not catalog all the uses for Position that a Market or Party may require.

7.1 Introduction

The purpose of the Position Facet is to allow access to the accumulated position for actors supporting specific Roles. A Party's ***Position*** for a time period is the algebraic sum of committed supply or sales for instruments overlapping that time period. A Party's position for an Instrument is computed from trades for that Instrument. In CTS, purchasing a Resource increases the Position, and Selling a Resource reduces the Position.

An Actor may, with appropriate authorization, request positions for other parties. This permits the specification and implementation of an auditor Actor. Roles using the Position Facet include:

- The Actor whose position is being requested—the *position Party.*
- An Actor who is authorized to request position information for other actors—including but not limited to an auditor—the *requestor.*

Position Interactions follow the Streams pattern. A request for position includes a bounding interval. The response reports, at least, the Position for each Interval included within the bounded Interval of the Request.

Table 7-1: Position Facet

Facet	Request Payload	Response Payload	Notes
Position	EiRequestPosition	EiReplyPosition	Request an Actor's Position(s) for a specific time interval and reply with those Position(s) if access is authorized.

This is the UML sequence diagram for the Position Facet:

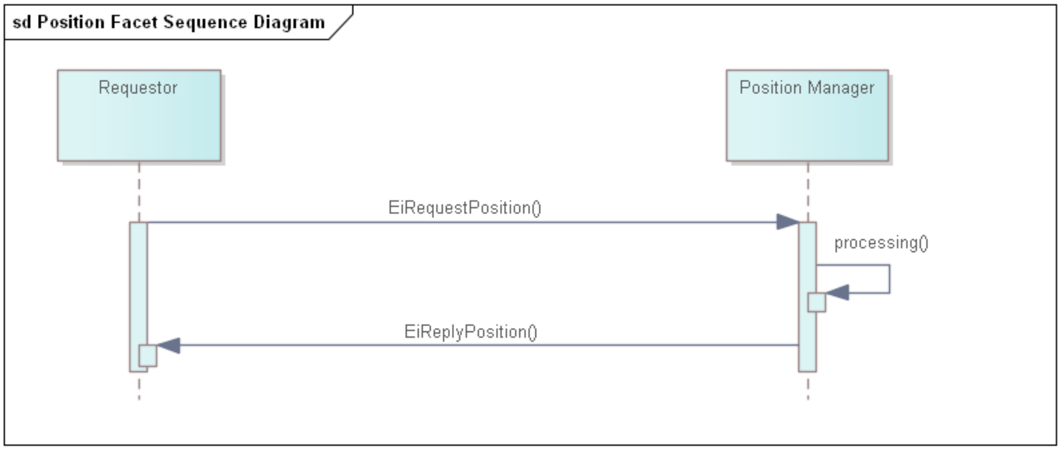


Figure 7-1: UML Sequence Diagram for the Position Facet

7.2 Information Model for the Position Facet

This Facet applies Section 3.3 *"The Bounding Interval Pattern in CTS".*

For Position, a bounding interval is specified and the position in each interval contained in the closed bounding interval is returned. A Request for Position specifies a Resource.

When the Position Request is for a Resource, then the Position is assembled from all Transactions for that Resource. When the Position Request is for a Product, then the Position is assembled from all Transactions for that Product. Consider, for example, a Position Request for Green Power, however defined, may only exist between 1:00 PM and 4:00 PM. The Position for Power for the rest of the day may be assembled from several sources, perhaps with different Warrants.

A Position is concerned with the total amount under contract, not the prices. If an Actor has positions in more than one Product, say, in a one-hour Product and in a one-minute Product, then the returned Position SHALL use the shorter Duration to denote the total amount.

The attributes are shown in the following section.

7.3 Payloads for the Position Facet

The Position payload is in the format of a CTS Stream, with only a Quantity in the Interval Payload. Position stated against the sum of Transactions in all Segments.

The [UML] class diagram describes the payloads for the Position facet.

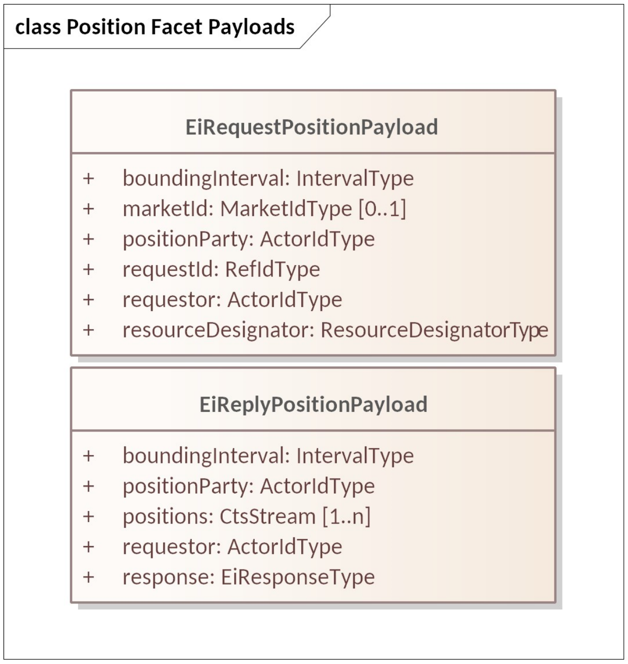

class Position Facet Payloads
EiRequestPositionPayload
+ boundingInterval: IntervalType
+ marketId: MarketIdType [0..1]
+ positionParty: ActorIdType
+ requestId: RefIdType
+ requestor: ActorIdType
+ resourceDesignator: ResourceDesignatorType
EiReplyPositionPayload
+ boundingInterval: IntervalType
+ positionParty: ActorIdType
+ positions: CtsStream [1..n]
+ requestor: ActorIdType
+ response: EiResponseType

Figure 7-2: UML Class Diagram of Payloads for the Position Facet

Table 7-2: Attributes of Position Facet Payloads

Attribute	Attribute Type	FIX Field	Meaning
Bounding Interval	Interval Type	Not in FIX	The [closed] time interval for which position information is requested. The first Positions Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the start of the Bounding Interval.
Market ID	Market ID Type	MarketID(1301)	Identifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13.
Position Party	Actor ID	PartyID (448)	The Party whose position is being requested. Allows a request for another Party's position, with appropriate privacy and security constraints
Resource Designator	Resource Designator Type	Not in FIX	The Resource for which Position is being requested. Should match the identified Market's Resource Designator
Request ID	Ref ID Type	PosReqID(710)	A reference to this payload. May be used as a correlation ID
Requestor	Actor ID	PartyID (448)	The Party requesting the position. A failure indication will be returned if the Requestor is not authorized to access position information for Position Party.

Moved down [3]: Not in FIX

Attribute	Attribute Type	FIX Field	Meaning
Positions	Cts Stream Type	Not in FIX	CTS Streams containing the positions for Position Party for each Resource. Positions are signed and may be zero. In CTS, purchasing a Resource increases the Position, and Selling a Resource reduces the Position. Each CtsStream interval that is contained within the Bounding Interval will have a value associated (signed integer). Note that a CtsStream contains a Resource Designator
Response	EI Response Type	Not in FIX	An EiResponse will indicate failure if Requestor is not authorized to access position information for Position Party for any of the requested intervals. See Section 2.4.

The purposes for requesting Position are system-specific and out of scope for this specification. Potential uses include:

- An Actor may request its own position(s) to recover from failure.
- A supplier of last resort may compare Positions to Delivery to impute transactions for unpurchased power delivered. (See Section 8 *The Delivery Facet").*

8 The Delivery Facet

The CTS Delivery Facet can be considered as the meter telemetry facet. We name it “Delivery” to align with the market focus of this specification, that is, a building takes delivery of power, or a distributed energy Resource (DER) delivers power. A CTS Delivery payload contains a CtsStream that conveys the measured or computed flow of a specific Resource through a particular point on the Resource’s delivery network during a specific Interval.

This Facet applies Section 3.3 “The Bounding Interval Pattern in CTS”.

CTS Delivery is typically derived from reading one or more meters, but it may be computed, implied or derived from some other method. Every Transaction is between a Party that promises to buy and a Party that promises to sell. Consider an actor that performs temporal arbitrage, i.e., buys one-hour Products and sells one-minute Products during the same hour. The Actor MAY report that it took delivery in each minute of that Interval, and the sales to other Actors MAY be visible only as reductions as recorded in Delivery.

In most cases, a node that takes delivery of more power or other Resource during an Interval than contracted for must eventually pay for that delivery. For example, An *auditor* (however defined) could sum all positions (See section 7, *The Position* Facet) and compare the result to total Delivery. The Auditor can then impute a transaction for the over-delivery. This may not be a simple “spot price”; if multiple Actors are taking over-delivery, then the last transaction is likely underpriced. Systems that track “actor reputation” may lower the reputation score. These examples explain the potential use of the information delivered by this facet and are not meant to suggest or dictate any particular business process or system model.

A CTS Delivery payload reports on the flow of a Resource and the duration of that report stream may not match the temporal granularity of any particular Product. The payload may (e.g.) include the sum of a one-hour market and of a one-minute market for the same Resource.

A CTS Market MAY assess penalties for Delivery outside certain bounds from the Position—as do many of today’s tariffed markets. Such bounds and penalties are out of scope for CTS. Computation and notification of Penalties is outside of scope.

A request for delivery specifies a Resource, unit of measure, bounding interval (Section 3.3), and a temporal granularity [Duration]. While the unit of measure and temporal granularity need to be within the capabilities of the telemetry node, they need not match any particular Product.

8.1 Interaction Pattern for the Delivery Facet

Table 8-1: Delivery Facet

Facet	Request Payload	Response Payload	Notes
Delivery	EiRequestDelivery	EiReplyDelivery	Request Delivery through a specific Measurement Point

Figure 8-1 is the UML sequence diagram for the Delivery Facet.

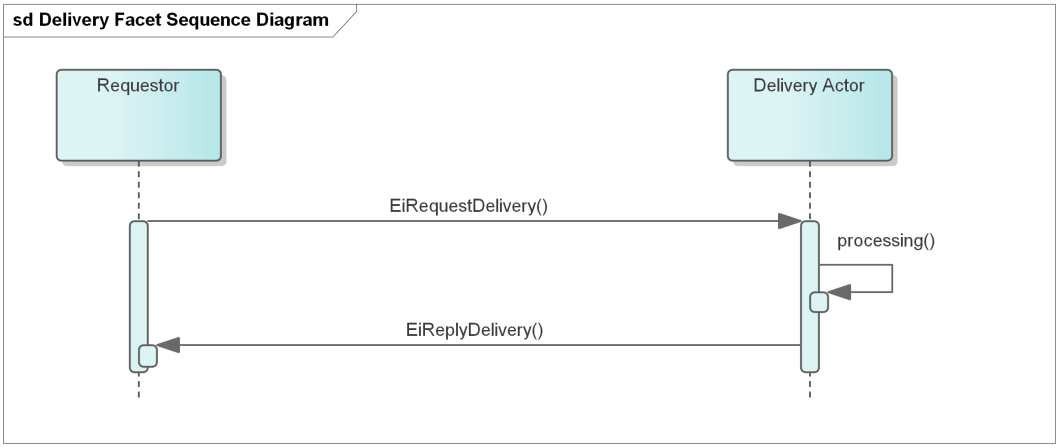


Figure 8-1: UML Sequence Diagram for the Delivery Facet

8.2 Information Model for the Delivery Facet

A Delivery response returns a single CtsStream of intervals of the requested Duration, with a quantity in each.

As with the Position Facet a bounding interval is specified and the delivery in each interval contained in the closed bounding interval is returned. The temporal granularity as requested may not be available, or the Delivery Actor may convert and combine—for example a request for one hour delivery intervals could be responded to using information from 1 minute or 5-minute measurement cycles.

The attributes are shown in the following section.

8.3 Payloads for the Delivery Facet

The **[UML]** class diagram describes the payloads for the Delivery facet.

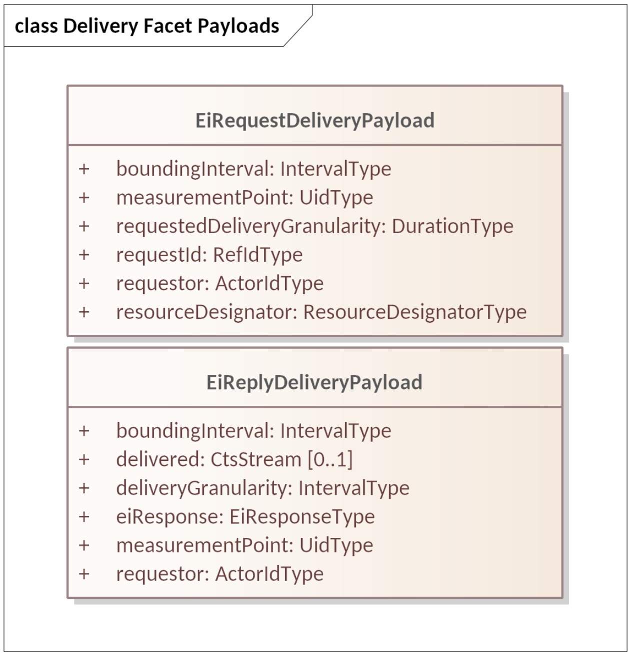

class Delivery Facet Payloads
EiRequestDeliveryPayload
+ boundingInterval: IntervalType
+ measurementPoint: UidType
+ requestedDeliveryGranularity: DurationType
+ requestId: RefIdType
+ requestor: ActorIdType
+ resourceDesignator: ResourceDesignatorType
EiReplyDeliveryPayload
+ boundingInterval: IntervalType
+ delivered: CtsStream [0..1]
+ deliveryGranularity: IntervalType
+ eiResponse: EiResponseType
+ measurementPoint: UidType
+ requestor: ActorIdType

Figure 8-2: UML Class Diagram of Payloads for the Delivery Facet

Table 8-2: Attributes of Delivery Facet Payloads

Attribute	Type	Meaning	Notes
Bounding Interval	Interval Type	The [closed] time interval for which position information is requested.	The first Delivered Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the end of the Bounding Interval. See Section.3.3 “The Bounding Interval Pattern in CTS”
Measurement Point	ID	An identification of the Point where the floe of the Resource is measured.	Information should be secure in conformance with appropriate privacy and security constraints
Requested Delivery Granularity	Duration Type	The granularity requested for delivery information	Temporal Granularity in reply, as in 1 hour. If empty, determined by capabilities of Measurement Point.
Request ID	Ref ID Type	A reference to this payload	May be used as a correlation ID
Requestor	Actor ID	The Party requesting the position.	Requestor must be authorized to access delivery information for this point. May be Party, auditor or other.
Delivered	CtsStream	A CtsStream containing the Quantity delivered in each Interval.	

Attribute	Type	Meaning	Notes
Response	EiResponse Type	An EiResponse. Will indicate failure if Requestor is not authorized to access information,	If the Requested Delivery Granularity cannot be used, the Response MAY indicate what granularit(ies) can be used. See Section 2.4.

Deleted: granularity

9 The Negotiation Facet

So far, this specification has described an order book market of simple Tender, Transaction and Delivery. This section discusses more advanced interactions. A Segment-based matching engine, however defined, matches Tenders to Buy and Tenders to Sell and creates Transactions.

With this Section, we introduce the messages used in Segments wherein the Buyer and the Seller jointly create matching Tenders. Negotiations rely on what FIX terms Pre-Trade Information. This section describes how Parties come to an agreement to create a Transaction through direct communication. The Parties conduct this conversation using requests for quotes, quotes, and quote responses. The Market facilitates the quote process but does not intervene—it acts as a neutral party.

Deleted: find

In essence, a Quote contains a Tender. The message accepting the Quote contains a matching Transaction and MUST reference the accepted Quote. The Parties must inform the Segment of the agreement, and the Segment processes the logical Transaction memorializing that agreement. The Market may still reject the agreement because of credit limits, or because a third Party has already lifted the quote (in case of public quotes), or because the Transaction would exceed operating limits of the system, or for some other reason.

Deleted: matching

Deleted: Tender.

Deleted: creates

Deleted: the Tenders are incompatible, or because

Deleted: ,

The messages and interactions are determined by the mechanism used in the market Segment. See Section 13 for a discussion of Market Mechanisms and how to select a Segment to trade in. Note: not all Markets must support all Market Mechanisms.

Requests for Quotes and Indicative Quotes (see below for definitions) may be public and if they are, they appear in a Quotes Ticker.

Financial markets assume that the same party, called the Quote Issuer, initiates all quotes in a specific negotiation. The recipient of a quote can accept the quote, if it is tradeable and the terms are agreeable, or reject the quote, i.e., end the negotiation. When a Party accepts ("hits" or "lifts") a tradeable quote, the Market executes the Transaction—the issuer of the quote cannot back out. A recipient MAY abandon the negotiation, choosing to initiate a new negotiation with a new Quote.

CTS negotiations differ from financial practice in that in financial negotiations, the instrument never changes. Over the course of a CTS negotiation, the time of delivery may change, which is a change of Instrument.

Negotiation may be used to enable large buyers to plan significant Resource use over time, for example, scheduling a long running industrial process which may also require off-market mechanisms such as labor planning. Such a buyer could submit multiple Requests for Quotes with different schedules, and then select from among the Quotes received in response.

This specification does not require that a Market include any of the scenarios described above. We include them to illustrate how the essential components of Negotiation might fit together in a specific market.

9.1 Negotiation Vocabulary

Negotiations use information elements defined above in TenderBase (5.3), also used in Tenders and Transactions. Note that the term Quote by itself includes both indicative and tradeable Quotes.

Table 9-1: Negotiation Terminology

Term	Purpose	Comment
Request for Quote (RFQ)	A Party submits a Request for Quote to try to find a market in an Instrument or Instruments. A Request for a Quote may be for a time range of Instruments.	May be used pre-opening to elicit tenders, both buy and sell, to determine market opening prices.

Term	Purpose	Comment
Quote	Indicates the price and quantity at which an instrument can be bought or sold. A Quote may be issued in response to an RFQ or it may initiate a negotiation.	The CTS Quote may be either a Bid Quote (buy) or an Ask Quote (sell). The initiator may choose to advertise any Quote to attract potential counterparties by requesting Publication.
Indicative Quote	A Quote that cannot be used to create a commitment leading to a Transaction.	As part of a Negotiation, a Party may submit a counter Quote to ask for a better Quote. Indicative quote(s) may also be issued in response to an RFQ.
Interval Quote	A Quote provided for only a Specific Interval.	Some Segments MAY limit negotiations to Intervals (in TenderBase) only by disallowing Streams. See Section 13.3, "Segment Reference Data"
Stream Quote	Prices and Quantities for a Product in a series of consecutive Instruments submitted as a single Quote.	In energy markets, a stream quote is often referred to as a "Load Curve."
Tradeable Quote	An offer to buy or sell up to a specific quantity of an Instrument for a specific price.	A Tradeable Quote is registered by the Segment and can be referenced ("lifted") to initiate a Trade as if it were a Tender.
Quote Response	A response to a an RFQ or Quote, The response may accept the Quote, or counter with another Quote or announce an end to a Negotiation.	Only a Tradeable Quote can be accepted to create a Transaction.
Private Quote Private RFQ	A quote or RFQ sent only to selected Counterparties during a Negotiation.	An implementation may use the Segment to distribute Quotes to Counterparties or it may expect Parties to message Counterparties directly.
Public Quote Public RFQ	A Quote or RFQ published to all subscribers to a Segment's Quotes Ticker. (See Section 11.5.2)	RFQs, Indicative Quotes, and Tradeable Quotes may be Published.
Issuer	The Issuer is the Party that originates a Quote, whether in response to an RFQ, or unsolicited.	The Issuer must accept a Transaction created in response to a Tradeable Quote in case of a positive Quote Response.

9.2 Narrative on Negotiation (non-normative)

An extended discussion of use cases and negotiation in markets is in Appendix B.

The Negotiation process is inherently flexible. A Transaction may come after many rounds of negotiation, or directly from a response to the first tradeable quote. This section describes some potential interactions to clarify the concepts before defining message types in the following sections.

A Party that wants to transact some quantity of a Resource may start a Negotiation by sending a Request for Quotation (RFQ) or an indicative Quote; a Tender simplifies the interaction

Message semantics and sequencing are in this section in the relevant diagrams and tables.

This Facet applies Section 3.3 "The Bounding Interval Pattern in CTS".

An RFQ uses an optional Bounding Interval to focus on what an acceptable response might be. The possible situations are.

Deleted: wishes

Deleted: amount

Deleted: resource, to find a potential counterparty, or to arrive at an agreement with a specific known counterparty…

Deleted: either

Deleted:),

Deleted: perhaps a

Deleted: .

Deleted: indicate

Deleted: would

(1) A Bounding Interval is included.[15]. This indicates that a Stream Quote that matches the Bounding Interval is likely to be acceptable. The responder has the option rejectinig the quote and starting a new negotiatiion by submitting a counter-quote, that is, initiating a new Quote/Response interaction, perhaps with a different Bounding Interval proposing a different Interval I[16].

Deleted: of

(2) A Bounding Interval is not included in the Quote or RFQ—any response must match the included interval or stream.[17]

RFQs and Quotes may be addressed directly to one or more potential counterparties or published to the entire Segment by means of a Ticker. The Market does not need to know about or register the RFQ or Indicative Quote because it cannot lead directly to a trade or Transaction. The recipient may issue a Quote Response to counter or reject the Quote. The recipient may also drop the Negotiation and start a new one by issuing a Quote or RFQ. See Section 13.1 "Market *Mechanisms*", as well as Appendix B for a discussion of interaction patterns in different markets.

Deleted: negotiation

When the Party that has received a Tradeable Quote decides that there is an essential meeting of requirements, a recipient accepts ("lifts") the Quote; in CTS, the recipient must inform the Market to create the Transaction using an EiAcceptQuote payload which in turn generates an EiAcceptQuote payload to each party.

Note that if multiple actors accept a quote, market serialization determines whiich accept is processed first.

Negotiations may include Interval Quotes or Stream Quotes, a pattern that matches that of Tenders (See Section 5.3.1, *"Interval Tenders and Stream Tenders."*) The stream in an RFQ need not fill the Bounding Interval; an overnight bounding interval of fifteen hours may be seeking any proposed three-hour stream during that interval.

9.3 Messages for the Negotiation Facet

A Request for Quotes (RFQ) is a message describing what is to be quoted, and may be sent to a Segment or to one or more intended counterparties.

A Quote is either unsolicited or in response to an RFQ. A recipient (CounterParty) of a Tradeable Quote may respond with a Quote response to accept it.

Deleted: spontaneous

[15] This is the same pattern used in Sections 3.3 "The Bounding Interval Pattern in CTS", 7 (Position Facet) and 8. (Delivery Facet)

[16] Consider a Buyer seeking a Seller willing to run a generator for three hours. The Seller, for economic or operational reasons is unwilling to run the generator for less than 6 hours, and returns a stream quote indicating this longer Interval.

[17] A CTS Stream may be considered to implicitly includes a bounding. Interval.

Table 9-2 Messages for the Negotiation Facet

Request Payload	Response Payload	Notes
EiCreateRfq	EiCreatedRfq	Create and send an RFQ. The RFQ is directed to intended Partys or published to the Segment. The sender of EiCreateRfq may request Publication , but has no guarantee that Publication occurs.
EiCancelRfq	EiCanceledRfq	Indicates that the RFQ Issuer no longer wishes to receive Quotes.
EiCreateQuote	EiCreatedQuote	Create and send a Quote. If the Quote is to be published, the Counterparty is the ID of the Market. Otherwise, it goes to the intended Counterparty. The sender of EiCreateQuote may request publication, but has no guarantee that the Market publishes the Quote.
EiAcceptQuote	EiAcceptedQuote	EiAcceptQuote is a specialization of EiCreateTransaction. EiAcceptedQuote as a specialization of EiCreatedTransaction returns any errors As with EiCreatedTransaction, the EiAcceptedQuote payload must be sent to Party and CounterParty.
EiCancelQuote	EiCanceledQuote	Cancel a Quote. This may be rejected if the Quote was tradeable and had already been lifted by the Counterparty.
EiRejectQuote	EiRejectedQuote	Recipient explicitly rejects referenced Quote.

Deleted: , and when successful returns a CreatedTransaction…

Deleted: ..

9.4 Interaction Pattern for the Negotiation Facet

These are the UML sequence diagrams for the Negotiation Facet. Different Market Mechanism Types (MMT) may have shortened Interaction patterns. Due to the complexity of the Quote diagram, we show the RFQ and Quote aspects in separate diagrams.

We have not shown the full Market-mediated interaction pattern iin which the Market initially passes messages through but needs to take action when agreement has been reached, by sending an EiAcceptQuote to both parties. See Section 9.4.3.

Where the Market mediates the interactions, in general the payloads are passed on. One exception is where the Market might add (e.g.) a Market Order ID to a payload. Other possible actions include posting a ticker payload.

9.4.1 Interaction Patterns for RFQ and Quote

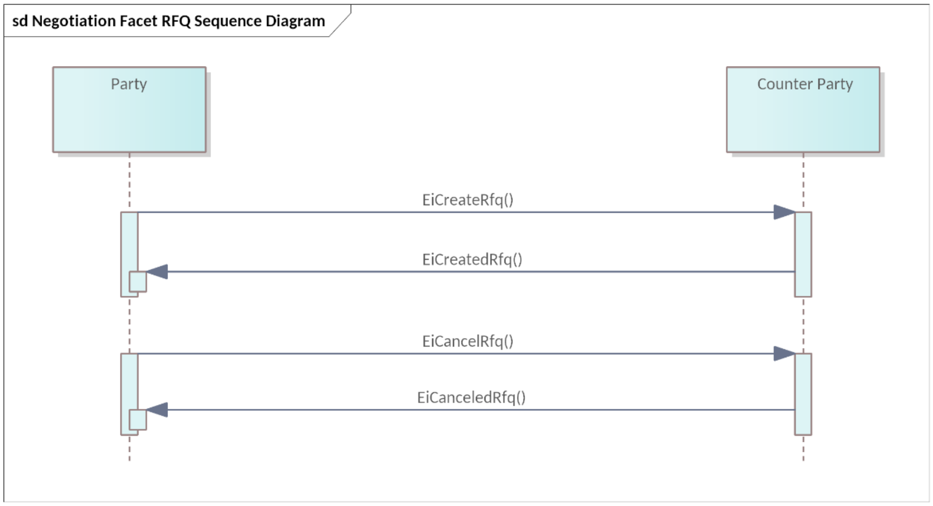


Figure 9-1 UML Sequence Diagram for Negotiation Facet RFQ (Request for Quote)

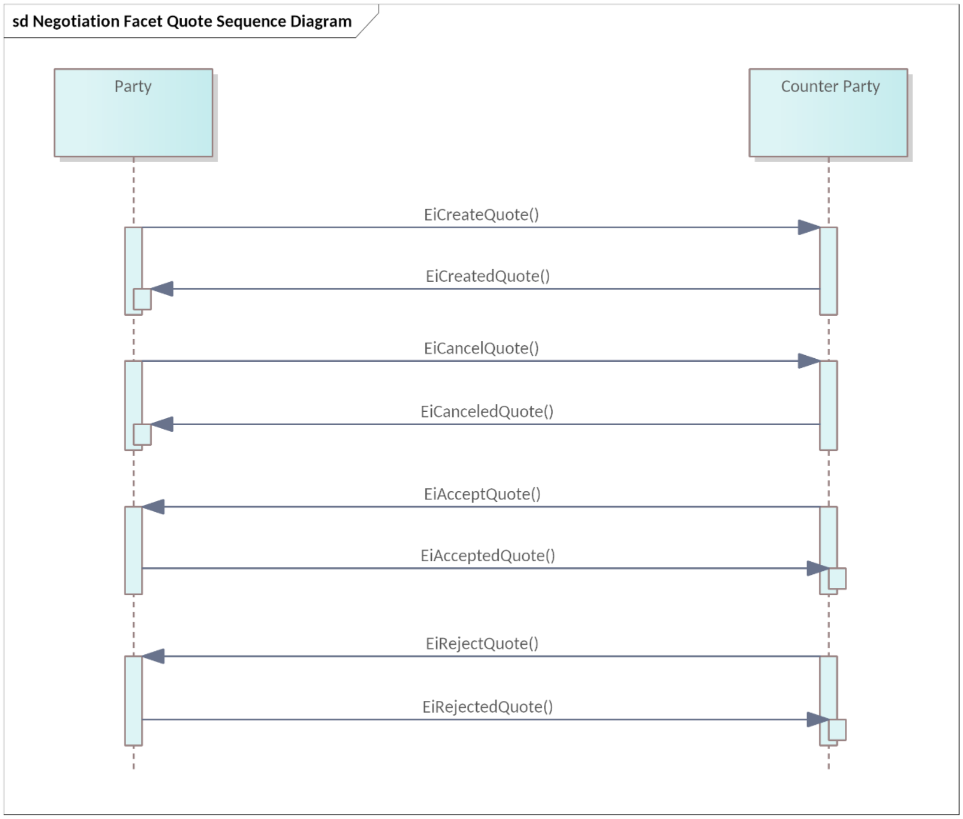


Figure 9-2 UML Sequence Diagram for Negotiation Facet Quote

9.4.2 Creating Transactions from Quotes

A Party receiving a Tradeable Quote MAY respond by submitting an AcceptQuote that references that Quote. The Market registers a Tradeable Quote it receives AS IF it were a Tender, and retains this information until it expires or is cancelled.

EiAcceptQuotePayload is a subclass[18] of an EiCreateTenderPayload that references the ID of the Tradeable Quote being accepted; see Section 6.4 for attributes. Figure 9-10 shows this relationship.

9.4.3 Interaction Pattern for Market-Facilitated Negotiation

Certain Quotes and RFQs may have attributes which require Market Segment knowledge. These include Private Quotes, Published Quotes, and Tradeable Quotes

As a partial example of this agency of a Market or Segment see Figure 9-3 “Market Mediated Quote and Responses Sequence Diagram”. Similar interaction patterns take place for other market-mediated interactions, e.g., for Tradeable Quotes.

[18] In UML formal terminology, EiAcceptQuotePayload and EiAcceptedQuotePayload are generalizations respectively of EiCreateTransactionPayload and EiCreatedTransationPayload. Informally, one would say “EiAcceptQuotePayload *is an* EiCreateTransactionPayload.” All attributes are inherited from the respective base class. See Figure 9-10 Negotiation Facet Accept and Accepted Quote Payloads.

Deleted: Mediated

Deleted: /

Deleted: EiCreateTenderPayload and EiCreatedTenderPayload.

Deleted: EiCreateTenderPayload.”…

Deleted: classes.

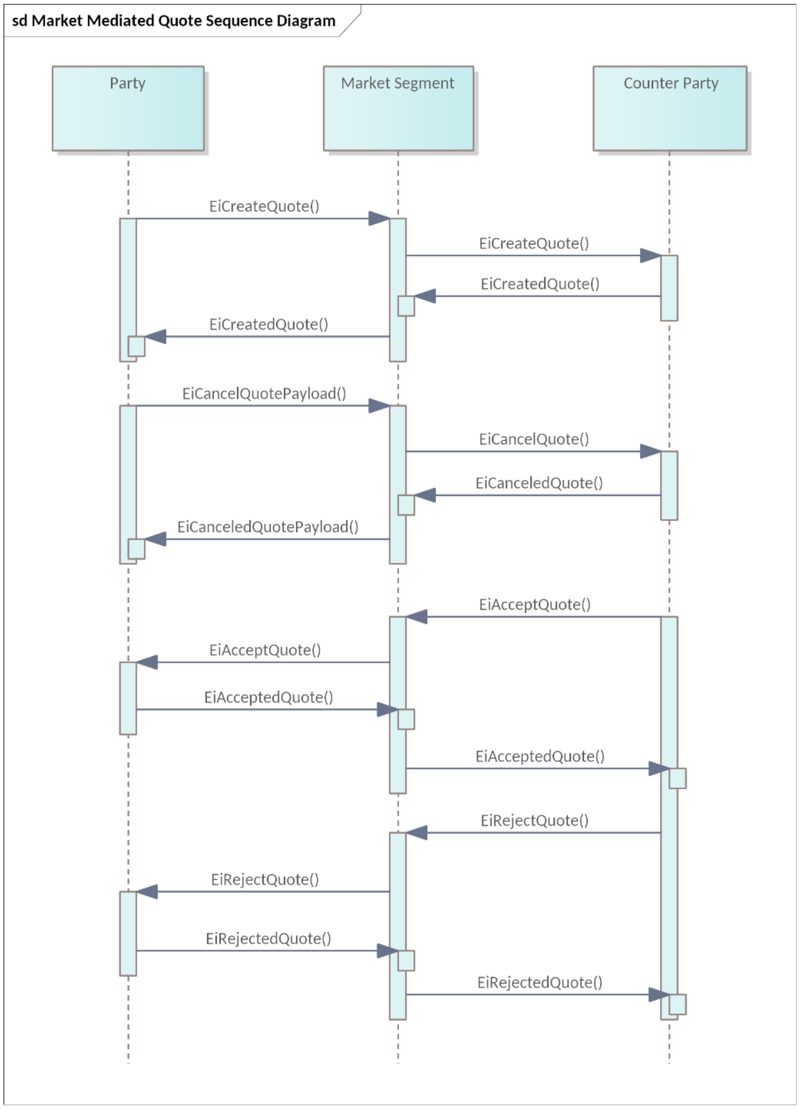
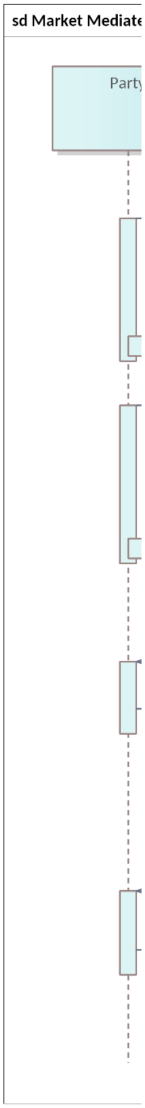


Figure 9-3 Market Mediated Quote and Responses Sequence Diagram

Deleted:



9.4.4 Interaction Patterns Restricted by Market Mechanism

Certain Market Mechanisms (See Section 13.1 *"Market Mechanisms")* restrict possible responses to an EiCreateQuote payload.

9.4.4.1 Quote-Driven Markets

Quote-Driven markets are typified by one or more dominant players who provide Quotes and Parties can lift some or all of each Quote. In a Quote-Driven Market Mechanism (MMT_QUOTE_DRIVEN) after receiving and responding to an EiCreateQuote the allowable responses after EiAcceptQuote are shown in the Sequence Diagram in Figure 9-4.

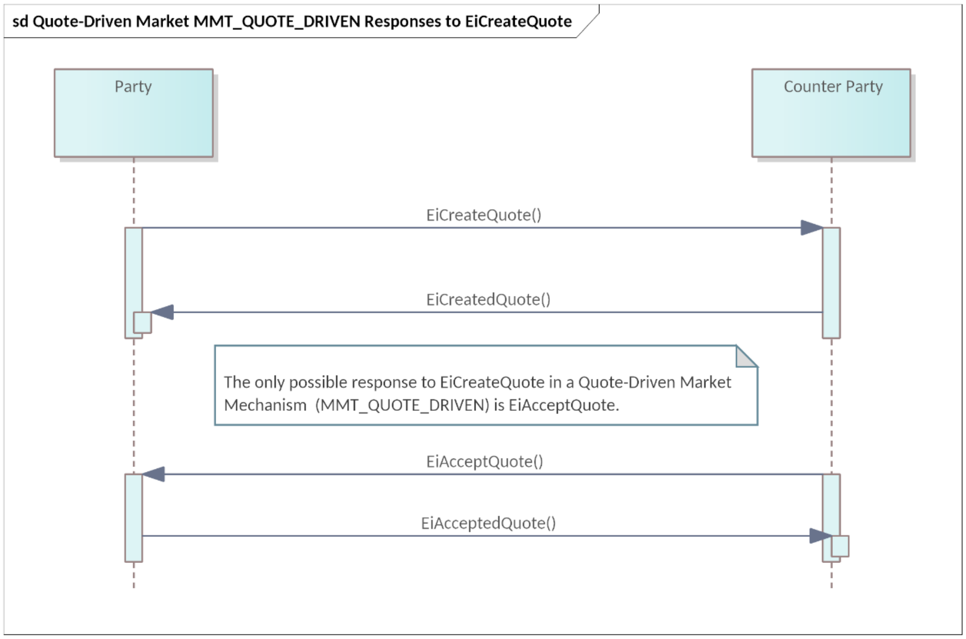

sd Quote-Driven Market MMT_QUOTE_DRIVEN Responses to EiCreateQuote
Party
Counter Party
EiCreateQuote()
EiCreatedQuote()
The only possible response to EiCreateQuote in a Quote-Driven Market Mechanism (MMT_QUOTE_DRIVEN) is EiAcceptQuote.
EiAcceptQuote()
EiAcceptedQuote()

Figure 9-4 Quote-Driven Market (MMT_QUOTE_DRIVEN) Responses to EiCreateQuote—the only possible response is EiAcceptQuote.

9.4.4.2 Request for Quotations Market

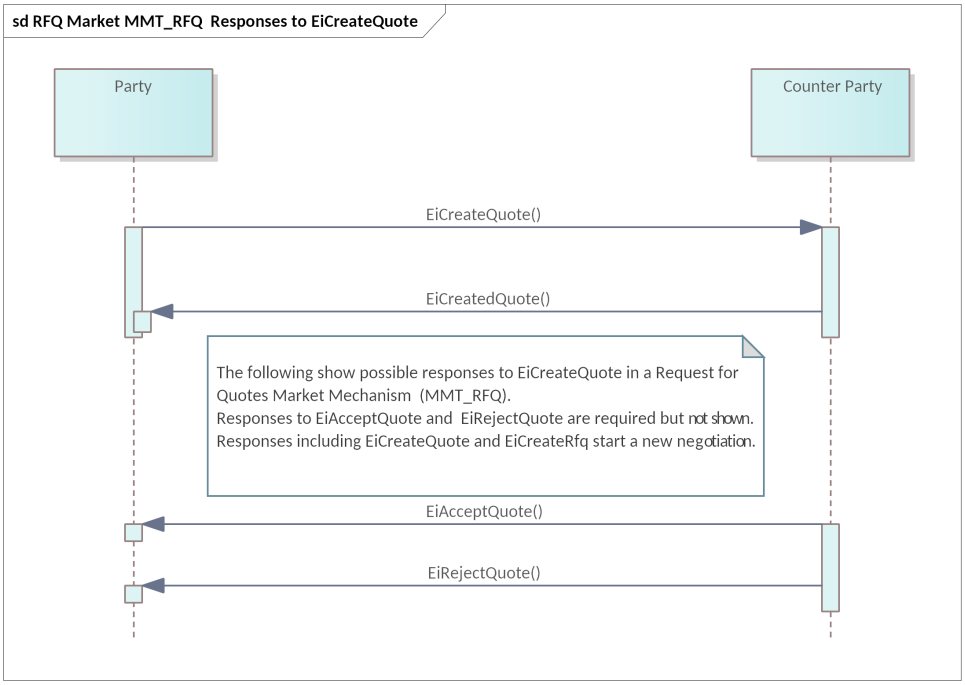
In a Request for Quotations Market (RQ) (MMT_RFQ) after receiving an EiCreateQuote and responding with EiCreatedQuote, the allowable responses are below and include EiCreateRfq as shown in the Sequence Diagram in Figure 9-5 “Request for Quotations Market (RQ) MMT_RFQ Responses to EiCreateQuote”.

- EIAcceptQuote
- EiRejectQuote

The following messages start a new negotiation and are not show in the figure:

- EiCreateQuote
- EiCreateRfq

Note that either EiCreateQuote or EiCreateRFQ initiates a new negotiation. A failure response stops the negotiiation in progress.



The following show possible responses to EiCreateQuote in a Request for Quotes Market Mechanism (MMT_RFQ).
Responses to EiAcceptQuote and EiRejectQuote are required but not shown.
Responses including EiCreateQuote and EiCreateRfq start a new negotiation.

Figure 9-5 Request for Quotations Market (RQ) MMT_RFQ Responses to EiCreateQuote are EiAcceptQuote and EiRejectQuote

9.5 Information Model for the Negotiation Facet

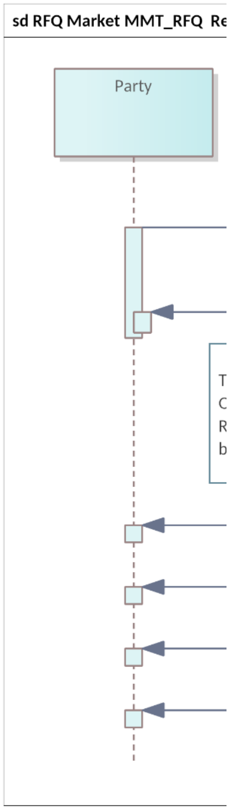
The RFQ can be considered as preliminary to a Quote, and so has more optionality.

9.5.1 A Note on Stream Quotes

Some Segments may permit or require Stream Quotes, that is, a single Quote for multiple consecutive Instruments.

Stream Quotes are treated as multi-legged tenders, that is, a Party that wishes to lift a Stream Quote must lift ALL of the Stream Quote. In Power markets, Stream Quotes are used to buy or to sell load

Deleted:



Deleted: *—responses to EiAccept, EiReject, EiCreateQuote and EiCreateRfq are required but not shown.…*

Deleted: ..

curves, that is, Power in each Interval over a longer time.[19]. Stream Quotes are generally used solely in Segments using a Request for Quotation mechanism. See Section 13.1 *"Market Mechanisms"* and Appendix B for discussions of Market Mechanisms.

Stream quotes and quotes are the same UML class and are related just as Interval Tenders and Stream Tenders are related.

9.5.2 The Request for Quotation

The RFQ can be considered as preliminary to a Quote, and so has more optionality. An RFQ could solicit a quote for 15 minutes of power sometime in an 8-hour window. It could be precise, as in a request for a specific amount of power for a specific duration at a specific time.

The UML Class Diagram for the EiRfqType are shown in Figure 9-6.

[19] A large generator may have a ramp up period to reach full power followed by a ramp down period. A long-running industrial process may issue RFQs to find the best time to run a process, and then lift a Quotation to select an operating schedule.

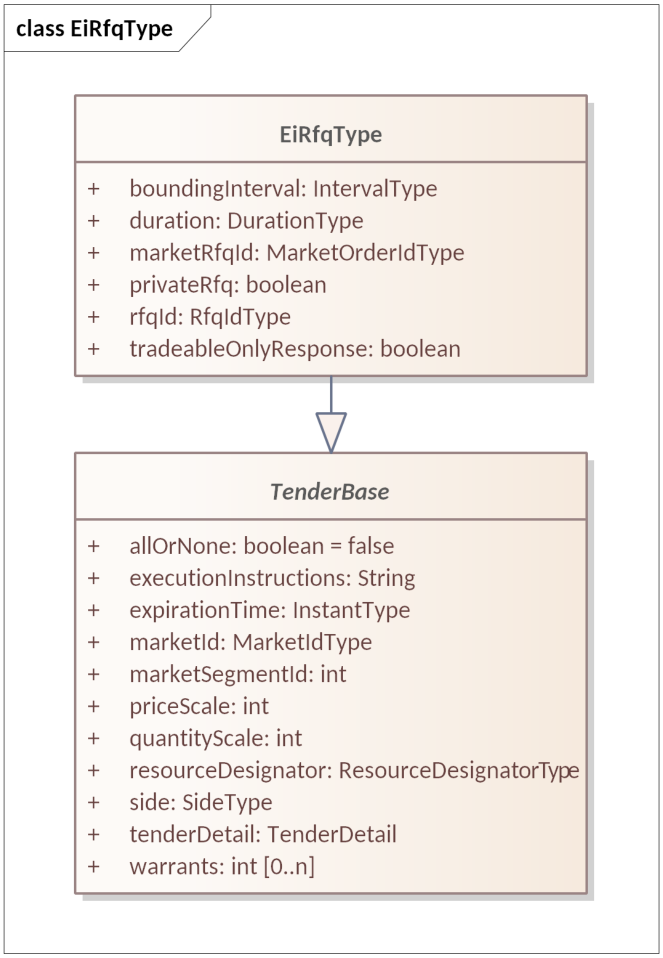

class EiRfqType
EiRfqType
+ boundingInterval: IntervalType
+ duration: DurationType
+ marketRfqId: MarketOrderIdType
+ privateRfq: boolean
+ rfqId: RfqIdType
+ tradeableOnlyResponse: boolean
TenderBase
+ allOrNone: boolean = false
+ executionInstructions: String
+ expirationTime: InstantType
+ marketId: MarketIdType
+ marketSegmentId: int
+ priceScale: int
+ quantityScale: int
+ resourceDesignator: ResourceDesignatorType
+ side: SideType
+ tenderDetail: TenderDetail
+ warrants: int [0..n]

Figure 9-6 UML Class Diagram for EiRfqType

Attributes of EiRfqType are shown in Table 9-3. Attributes inherited from TenderBase are defined in Table 5-3.

For example, consider a Party requesting a Quote for three hours of Power this evening between 4 PM and midnight. The RFQ would have a Bounding Interval 4 PM to 12M and a Duration of 3 hours of the amounts specified in the TenderBase. If the TenderBase references a Segment of 0, the Request goes to all Segments.

Table 9-3 Attributes of EiRfqType

Attribute	Type	FIX Field	Meaning	Notes
Bounding Interval	Interval Type	Not in FIX	The [closed] time interval for which information is requested.	A Quote outside the Interval is permitted. In the example above, this is the “4pm to Midnight”. See Section 3.3 “The Bounding Interval Pattern in CTS”

Attribute	Type	FIX Field	Meaning	Notes
Duration	Duration Type	Not in FIX	The desired duration in the responding responsive Quote.	This is the “3 hours” in the example above. Zero means not specified.
Market RFQ ID	Market Order Id Type	Not in FIX	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages In FIX the Market does not issue its own IDs when it merely supports the negotiation between the parties with its infrastructure.In contrast, CTS requires a Market-assigned ID, which requires the Market to create and attach those IDs.
Private RFQ	Boolean	PrivateQuote (1171)	The RFQ is specific to a single Party.	
RFQ ID	RFQ ID Type	QuoteReqID(131)	ID assigned by originating Party.	
Tradeable - Only Response	Boolean	QuoteType(537)	Indicates whether the initiator wants only Tradeable Quotes in response.	
Attributes for TenderBase are defined in Table 5-3 *Tender Base Attributes*				

Moved (insertion) [4]

Deleted: OrderID (37)

Deleted: Related to

Deleted: RefOrderID (1080)

Deleted: See also FIX ClOrdID (11)

Moved up [4]: Not in FIX

9.5.3 Quotes

As described in Section 5.3 *“Information Model for the Tender Facet”* and in this Section, EiRfq and EiQuote are subclasses of and inherit from abstract class TenderBase. In Table 9-4, only the first five attributes are part of EiQuoteType; the rest are inherited as shown.

Figure 9-7 is a UML Class Diagram of EiQuoteType showing inherited and included attributes.

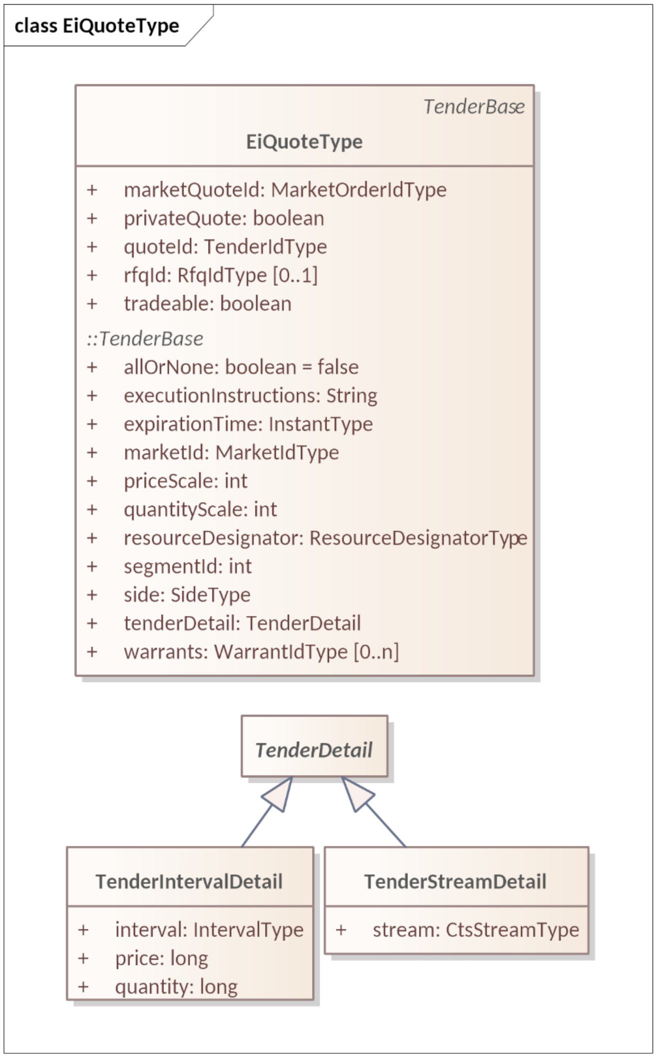


Figure 9-7 UML Class Diagram of EiQuoteType showing inherited attributes.

Deleted:

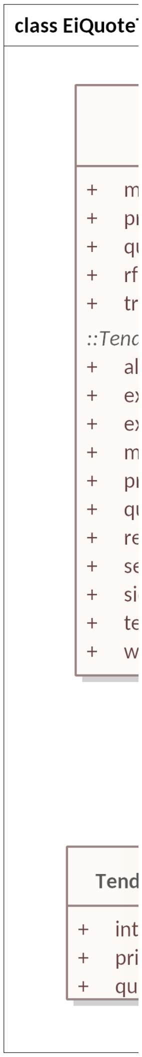


Table 9-4 Attributes of EiQuoteType

Attribute	Type	FIX Field	Meaning	Notes
Market Quote ID	Market Order ID Type	Not inFIX	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages.
Private Quote	Boolean	Private Quote (1171)	Quote is available specified counterparty only.	Quote is not available to the Segment.

Deleted: OrderID (37)

Attribute	Type	FIX Field	Meaning	Notes
Quote ID	Tender ID Type	QuoteID(117)	ID as submitted by Quote originator/issuer	Used in off-market negotiation
RFQ ID	RFQ ID Type	QuoteReqID (131) is related	Market-assigned ID of the RFQ to which this quote is responding	Referenced by a Quote responding to RFQ. Optional. In EiRfqType both a requester and market-assigned IDs are present.
Tradeable	Boolean	QuoteType (537)	Indicates whether the Quote is tradeable or not	If true, the quote is tradeable. If false the quote is not tradeable, which is by definition an Indicative Quote, consistent with FIX terminology.
Attributes for TenderBase are defined in Table 5-3 *Tender Base Attributes*				

Deleted: All Other

Deleted: as

Deleted: TenderBase

The Quote, RFQ, and Tender share common information using TenderBase. See Figure 5-2 *"UML Class Diagram Showing Commonality between Tender, Quote, and RFQ".*

9.6 Messages for the Negotiation Facet

9.6.1 RFQ Messages

The UML Class Diagram for the RFQ payloads is shown in Figure 9-8 below.

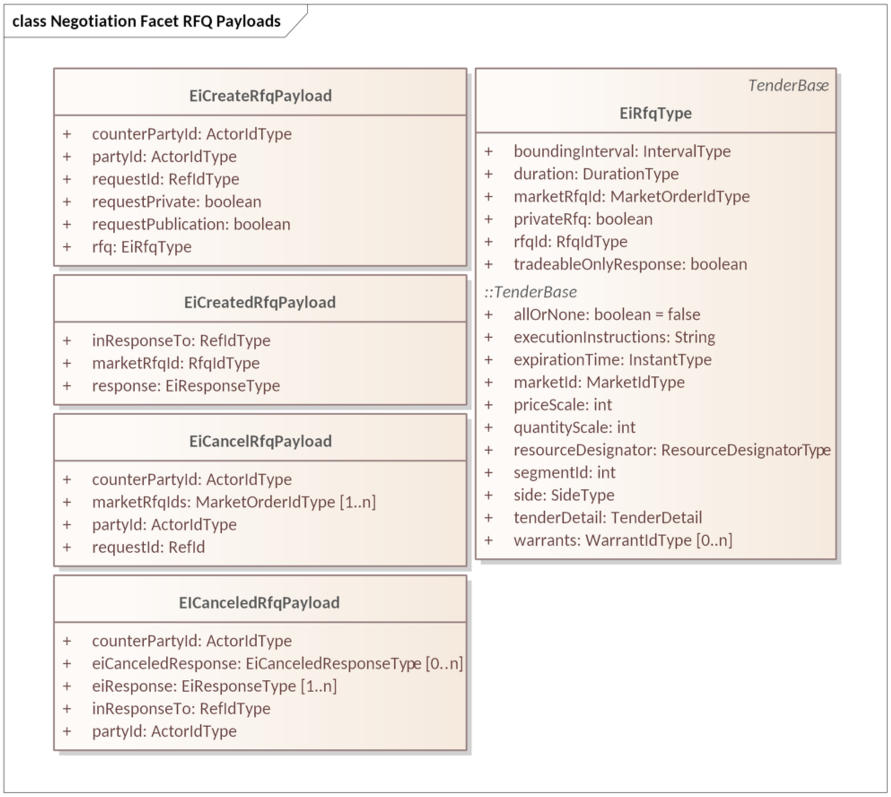

class Negotiation Facet RFQ Payloads
EiCreateRfqPayload
+ counterPartyId: ActorIdType
+ partyId: ActorIdType
+ requestId: RefIdType
+ requestPrivate: boolean
+ requestPublication: boolean
+ rfq: EiRfqType
EiCreatedRfqPayload
+ inResponseTo: RefIdType
+ marketRfqId: RfqIdType
+ response: EiResponseType
EiCancelRfqPayload
+ counterPartyId: ActorIdType
+ marketRfqIds: MarketOrderIdType [1..n]
+ partyId: ActorIdType
+ requestId: RefId
EICanceledRfqPayload
+ counterPartyId: ActorIdType
+ eiCanceledResponse: EiCanceledResponseType [0..n]
+ eiResponse: EiResponseType [1..n]
+ inResponseTo: RefIdType
+ partyId: ActorIdType
TenderBase
EiRfqType
+ boundingInterval: IntervalType
+ duration: DurationType
+ marketRfqId: MarketOrderIdType
+ privateRfq: boolean
+ rfqId: RfqIdType
+ tradeableOnlyResponse: boolean
::TenderBase
+ allOrNone: boolean = false
+ executionInstructions: String
+ expirationTime: InstantType
+ marketId: MarketIdType
+ priceScale: int
+ quantityScale: int
+ resourceDesignator: ResourceDesignatorType
+ segmentId: int
+ side: SideType
+ tenderDetail: TenderDetail
+ warrants: WarrantIdType [0..n]

Figure 9-8 UML Class Diagram Showing Negotiation Facet RFQ Payloads

The attributes of EiCreatedRFQ response payloads are in Table 9-5.

Deleted:

Table 9-5: EiCreateRFQ Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID Type	PartyID (448)	The Party IDs for the CounterParties for which the RFQ is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counter-party is used.
Party ID	Actor ID Type	PartyID (448)	The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side EiCreateTender.
Request ID	RefIDType	Not in FIX	Reference to this message payload	
Request Private	Boolean	PrivateQuote (1171)	The sender requests that RFQ to be Private only to specified Counter Party or Parties.	FIX has Public as an antonym of Private; due to privacy related market rules, CTS separates the concepts and clarifies that it is a request.
Request Publication	Boolean	PrivateQuote (1171)	Publication of the RFQ is requested.	The sender of an EiCreateRfq Payload requests publication on the Quotes Ticker if available. This is a request and may not take place. See also Request Private.
RFQ	EiRfqType		The RFQ transmitted.	An RFQ may use a Stream to indicate what sort of Stream Quote it is looking for or even multiple Streams to indicate an interest in transactions over time.
Fields of EIRfqType are shown in Figure 9-6 and Table 9-3				

Moved (insertion) [5]

The attributes for ECancelRfq and EiCanceledRfq are in Table 9-6.

Table 9-6 EiCancelRfq and EiCanceledRfq Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the Counterparty for which the Tender is created.	Unlike Tenders, Negotiations are typically directed to a specific Party. If the Quote or RFQ is published, the Counterparty is the Party ID of the Segment.
Market Request for Quote ID	Market Order Id	Not in FIX	Market-assigned ID for Request for Quote	Market Assigned in parallel with Tenders.

Moved (insertion) [6]

Deleted: OrderID (37)

Attribute	Type	FIX Field	Meaning	Notes
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this RFQ is made.	Indicates which Actor proposes the buy or sell side RFQ.
Request ID	Ref ID	QuoteReq ID(131)	An identifier for this Cancel RFQ Payload	
Ei Canceled Response	EiCanceled Response Type		Optional Detailed response for each RFQ for which cancelation was requested	
In Response To	RefIdType		The EiCancelRfqPayload that is responded to in the Canceled Payload	

9.6.2 Quote Messages

The UML Class Diagram for the Quote payloads is shown below. Attributes are in tables starting with Table 9-7.

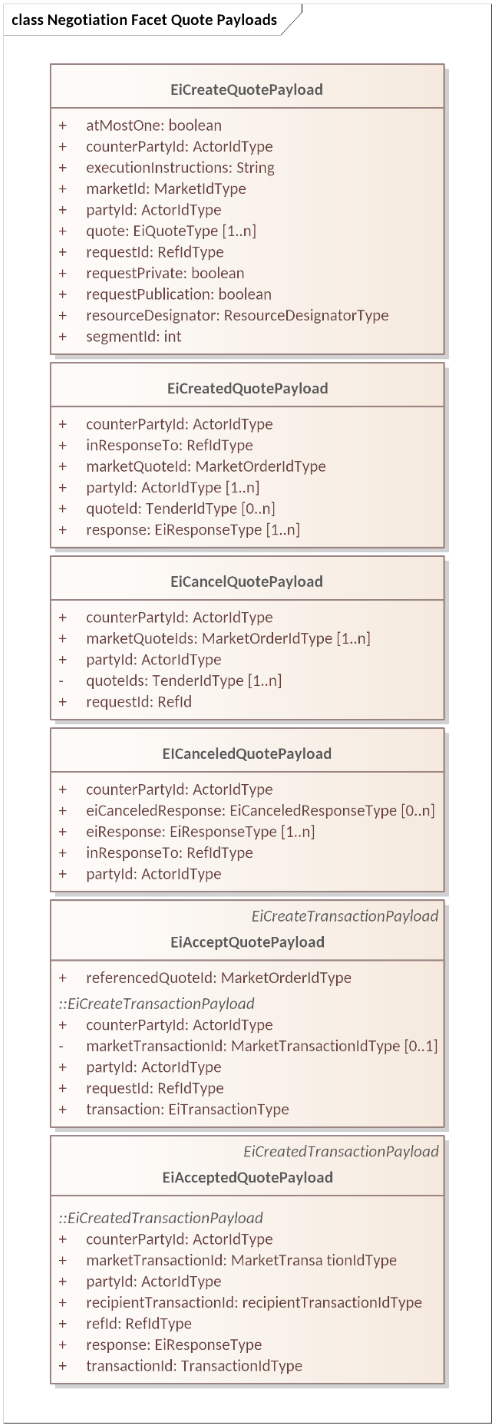

class Negotiation Facet Quote Payloads
EiCreateQuotePayload
+ atMostOne: boolean
+ counterPartyId: ActorIdType
+ executionInstructions: String
+ marketId: MarketIdType
+ partyId: ActorIdType
+ quote: EiQuoteType [1..n]
+ requestId: RefIdType
+ requestPrivate: boolean
+ requestPublication: boolean
+ resourceDesignator: ResourceDesignatorType
+ segmentId: int
EiCreatedQuotePayload
+ counterPartyId: ActorIdType
+ inResponseTo: RefIdType
+ marketQuoteId: MarketOrderIdType
+ partyId: ActorIdType [1..n]
+ quoteId: TenderIdType [0..n]
+ response: EiResponseType [1..n]
EiCancelQuotePayload
+ counterPartyId: ActorIdType
+ marketQuoteIds: MarketOrderIdType [1..n]
+ partyId: ActorIdType
- quoteIds: TenderIdType [1..n]
+ requestId: RefId
EiCanceledQuotePayload
+ counterPartyId: ActorIdType
+ eiCanceledResponse: EiCanceledResponseType [0..n]
+ eiResponse: EiResponseType [1..n]
+ inResponseTo: RefIdType
+ partyId: ActorIdType
EiCreateTransactionPayload
EiAcceptQuotePayload
+ referencedQuoteId: MarketOrderIdType
::EiCreateTransactionPayload
+ counterPartyId: ActorIdType
- marketTransactionId: MarketTransactionIdType [0..1]
+ partyId: ActorIdType
+ requestId: RefIdType
+ transaction: EiTransactionType
EiCreatedTransactionPayload
EiAcceptedQuotePayload
::EiCreatedTransactionPayload
+ counterPartyId: ActorIdType
+ marketTransactionId: MarketTransa tionIdType
+ partyId: ActorIdType
+ recipientTransactionId: recipientTransactionIdType
+ refId: RefIdType
+ response: EiResponseType
+ transactionId: TransactionIdType

Figure 9-9 Negotiation Facet Quote Payloads

The following tables show attributes for the Quote Payloads.

Deleted: Messages

Table 9-7 EiCreateQuotePayload

Attribute	Type	FIX Field	Meaning	Notes
At Most One	Boolean	ContingencyType (1385)	Used to express alternatives, only one of which is to be effective	See Trading Instructions in Table 5-4. First match cancels other Tenders.
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Quote is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Execution Instructions	String	ExecInst(18)	Execution Instruction.	Used only for multi-leg, and applies to all tenders in multi-leg. Execution instructions apply to each Tender in the List.
Market ID	Market ID Type	MarketID(1301)	. Market ID	Identifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13
Segment ID	Integer	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side
Quote	EiQuote Type		The quote transmitted by this message payload	One or more quotes
Request ID	RefIDType	QuoteMsgID(1166)	Reference to this message payload	
Request Publication	Boolean	PreTradeAnonymity(1091)		The sender of EiCreateQuote (the initiator) requests publication by setting Request Publication to true. This is a request—there is no guarantee that publication is performed.

Deleted: See

Table 9-8 EiCreatedQuotePayload

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Quote is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counter-party is used.
In Response To	Ref ID	QuoteMsgID (1166)	An identifier for the payload to which this is a response	
Market Quote ID	Market Order ID Type	SecondaryQuoteID(1751)	ID for this quote assigned by the Segment or Market	Used in acknowledgement and in future market messages
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side
Quote ID	Tender ID Type	QuoteID(117)	The quote transmitted by the EICreateQuote message payload	Zero or more quotes
Response	EiResponse Type	QuoteAckStatus (1865) QuoteReject Reason(300)	Specific error responses	See Section 2.4

Deleted: Not in FIX

Deleted: OrderID (37)

Deleted: OrderID

The Segment normally does not publish a Quote that is private or directed to a specific Party or Parties. If the Quote Issuer requests Publication, then the Segment MAY do so following its anonymization and publication practices. A Segment Publishes a Quote by distributing it using the Quotes Ticker. See Section 11.5.2, *"Quote Ticker"*

While the Quote Issuer can request Publication, the decision to Publish a Quote is made by the Segment. The Segment MAY be required to Publish Quotes from Parties identified as significant in the Market. Another Segment may decline to publish any Quotes to comply with privacy regulations.

9.6.2.1 Cancelling a Quote

A Party May cancel a Quote at any time so long as it has not previously been lifted by a Counterparty. The attributes of the Ei Cancel Quote payloads are in Table 9-9 and Table 9-10.

Table 9-9 EiCancelQuote Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Market Quote IDs	Market Order ID Type	SecondaryQuoteID(1751)	ID assigned by the Segment or Market.	One or more Market Quote IDs to request cancelation.

Deleted: OrderID (37)

Party ID	Actor ID	PartyID (448)	Actor ID for the Party that created the Tender	
Request ID	Ref ID	QuoteReqID (131)	An identifier for this Cancel Tender Payload	

Moved up [5]: Not in FIX

Table 9-10 EiCanceledQuote Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market
Ei Canceled Response	Canceled Response Type	Not in FIX	Detailed response for each quote that was included in the EiCancelQuote Payload	
EiResponse	EiResponse Type	QuoteAckStatus (1865) QuoteReject Reason(300)	Specific error responses	See Section 2.4
In Response To	Ref ID	QuoteMsgID(1166)	An identifier for the Cancel Tender Payload to which this is a response	
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender was made.	Indicates which Actor proposes the buy or sell side EiCreateTender.

Moved up [6]: Not in FIX

Moved down [7]: Not in FIX

9.6.2.2 Accepting a Quote

To accept a Tradeable Quote, whether on first notice or after negotiation, a Party submits an EiAcceptQuote Payload matching the Price and Quantity of the Quote and referencing the Market Quote ID. FIX and financial markets call this *lifting a quote*.

The EiAcceptQuote payload is exactly an EiCreateTransaction payload with the addition of a reference to the quote accepted because the match has been performed by the Quote/Accept cycles.

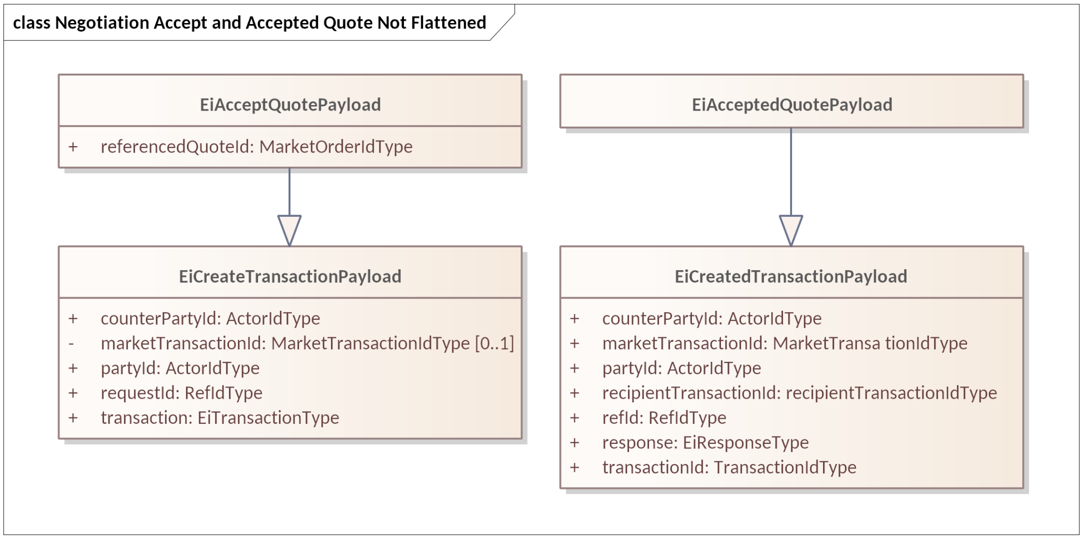

class Negotiation Accept and Accepted Quote Not Flattened
EiAcceptQuotePayload
+ referencedQuoteId: MarketOrderIdType
EiAcceptedQuotePayload
EiCreateTransactionPayload
+ counterPartyId: ActorIdType
- marketTransactionId: MarketTransactionIdType [0..1]
+ partyId: ActorIdType
+ requestId: RefIdType
+ transaction: EiTransactionType
EiCreatedTransactionPayload
+ counterPartyId: ActorIdType
+ marketTransactionId: MarketTransa tionIdType
+ partyId: ActorIdType
+ recipientTransactionId: recipientTransactionIdType
+ refId: RefIdType
+ response: EiResponseType
+ transactionId: TransactionIdType

Figure 9-10 Negotiation Facet Accept and Accepted Quote Payloads

The Market will then validate the accepted Quote and create a Transaction if it fits by sending an EiCreateTransaction Payload. The TenderBase in the EiAcceptQuote must match Instrument, Price, and Quantity in the Quote, except in a Quote-Driven Market, wherein EiAcceptQuote can lift a part of the Quote. Quotes in Segments with Market Mechanisms other than Quote Driven must have an execution instruction of All-or-None. The Segment typically maintains the balance remaining in a Quote.

Deleted: match

Notwithstanding any negotiation, the Market may reject the Accept Quote if accepting it would interfere with resource operations or violate financial requirements on participants.

If a Tradeable Quote is open when the Instrument closes, it is the responsibility of the Party that submitted the Quote to cancel it, and/or to have an appropriate expiration for the Quote. If the issuer still wishes to accept an instrument scheduled for 11:00 at 11:30, that is up to the Parties; how that is accomplished is out of scope of CTS. The Market will enforce its own rules for accepting the Transaction.

9.6.2.3 Rejecting a Quote

When a Party wants to end further negotiation, it replies with a Reject Quote message. Reject Quote iis in the choreography for an RFQ Market (MMT_RFQ) but not iiini a Quote-Driven Market.

The UML class diagram for EiRejectQuote and EiRejectedQuote is in Figure 9-11 Attributes are described in Table 9-11.

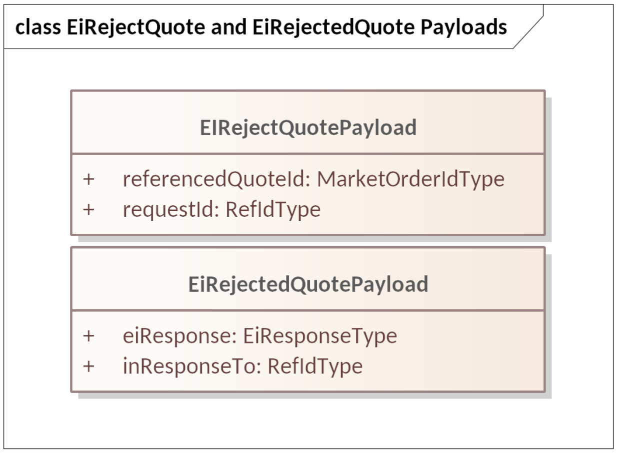

class EiRejectQuote and EiRejectedQuote Payloads
EIRejectQuotePayload
+ referencedQuoteId: MarketOrderIdType
+ requestId: RefIdType
EiRejectedQuotePayload
+ eiResponse: EiResponseType
+ inResponseTo: RefIdType

Figure 9-11 EiReject and EiRejectedQuote Payloads

Table 9-11 EiReject and EiRejected Quote Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Referenced Quote ID	Market Order ID Type	QuoteID (117)	The Market-assigned ID for the quote being rejected.	
Request ID	RefIDType	QuoteMsgID(1166)	Reference to this message payload	
Response	EiResponse Type	Not in FIX	Specific error responses	See Section 2.4
In Response To	Ref ID	QuoteMsgID(1166)	An identifier for the Cancel Tender Payload to which this is a response	

Deleted: PartyID
(448

Deleted: Actor

Deleted: In CTS, generally the PartyID for the Market

Deleted: CounterParty for which the Tender is created

Moved down [8]: Not in FIX

Deleted: EiResponse

Moved down [9]: Not in FIX

10 Subscription Facet

A Party wishing to trade in a market naturally wants to be kept apprised of changing information about the market. This can be roughly divided into granular information about what other Parties are doing in the Market, and information about the Market, Segment, Trading Session, or instruments (high price, low price, quantity sold, etc.).

In this section we describe the common aspects of subscriptions, including starting and stopping, or a one-time information message.

The FIX Protocol specification describes these as Market Data, that is, granular or aggregate information about activities in a Market, and Market Structure Reference Data, that is, information about how each Market Segment is operating.

FIX distinguishes between

- *Reference Data* which changes very slowly if at all—think the name of a market, or that a market segment trades one hour energy, and
- *Dynamic Data* which changes more frequently—think orders to buy or sell an instrument, session trading status, session intraday unscheduled auctions, and the like.

In the FIX Protocol, a Party gets this information either by means of Subscriptions or request messages resulting in a single response message. A Party subscribes to the information it needs and thereafter receives periodic updates relating to that subscription. The FIX interaction model defines a *subscription* as how an Actor requests one or more market reports.

A Market consists of multiple Market Segments, each trading a single Product based on the Resource traded in that market. Multiple Market Segments in a Market MAY trade the same Product, perhaps with different trading rules, or different schedules of operation. The Segments in a Market may support different Market Structure Reference Data reports. Information about a Market and its Segments is conveyed in the Market Structure Reference Data Subscriptions.

Subscriptions are how a Party requests specific Pre-Trade information. Not all Markets and Segments will support all Subscription types. The Subscriptions supported by each Segment are described in Section 13, *“Market Structure Reference* Data: Market, Segment, and Session Subscriptions*”.*

The following sections each use the Subscription pattern:

- Section 11, *“*Tickers”—Tenders, Quotes, RFQs, and Transactions
- Section 12, “Instrument Data Subscriptions”—outstanding tenders to buy or sell, high and low prices
- Section 13, “Market Structure Reference Data: Market, Segment, and Session Subscriptions*”*—slowly or unchanging reference data, more changeable dynamic data

Some markets or segments may not support fine-grained subscriptions. In such cases, the Managed Subscription payload and/or Market Structure Data Report MAY indicate a multi-cast point or other source to which an actor may choose to listen.

In CTS, the message transport is layered and out of scope.

10.1 Messages for the Subscription Facet

All subscriptions follow a common pattern for creation, management, and cancelation. This facet includes messages for Tickers, Instrument Data, and Market, Segment, and Session Data, as described in the following sections. Those messages inherit from the core subscription messages, which are of abstract type as no actual messages use only this base.

Table 10-1 Messages for the Subscription Facet

Facet	Request Payload	Response Payload	Notes
Subscription	EiManageSubscription	EiManagedSubscription	Create, manage, and cancel subscriptions

10.2 Interaction Pattern for the Subscription Facet

There is no UML sequence diagram for the Subscription Facet because the payload is abstract. The manage interactions are defined in Sections 11, 12, and 13. Specific subscriptions inherit from this pattern.

10.3 Information Model for Subscription Requests

The UML Class Diagram for the Subscription Request and Response is shown in Figure 10-1. Specific requests for tickers, instrument, and market information are defined in the following sections.

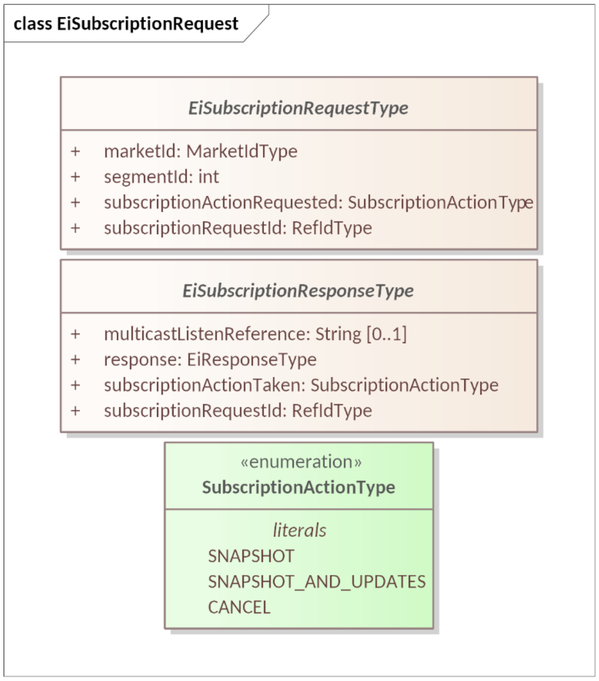

class EiSubscriptionRequest
EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL

Figure 10-1 UML Class Diagram for Subscription Request and Response Types

Attributes for the Subscription Request are shown in Table 10-2. We follow FIX's approach in using the message ID (RefID) as the subscription identifier.

Table 10-2 EiSubscriptionRequest Attributes

Attribute	Attribute Type	FIX Field	Meaning
Market ID	Market ID Type	MarketID(1301)	Identifier of the market of interest. An actor MAY be able to participate in more than one Market. See Section 13 “Market Structure Reference Data: Market, Segment, and Session Subscriptions”
Segment ID	Integer	MarketSegmentID(1300)	The FIX MarketSegmentID is a UID represented by a string; CTS uses an integer for Segment ID. If Segment ID is non-zero, the request is limited to reporting on the indicated single Segment. If zero, the subscription requests reporting on all Segments of the Market.
Subscription Action Requested	Subscription Action Type (enumeration)	SubscriptionRequestType (263)	The Subscription response type requested. CTS uses an enumeration that matches the pattern of FIX numeric codes: 0 – **SNAPSHOT** 1 – **SNAPSHOT_AND_UPDATES** 2 – **CANCEL** See the discussion following this Table.
Subscription Request ID	Ref ID Type	MDReqID(262)	Used to identify this request for managing a subscription. This is an identifier for the subscription and must be used to cancel. See ALSO FIX MarketDataRequest (35=DR).

Attributes for Subscription Response are shown in Table 10-3.

Subscriptions are inherently asynchronous. A Snapshot subscription request asks for a full report when the provider responds. A Snapshot and Updates subscription returns a full report and will return regular updates at some future times. A Cancel subscription stops all future Updates and ends the Subscription.

There is no expectation that each market participant can or should be able to get perfect knowledge about the trading behavior of all other participants; and creating a capability of doing so would likely prevent the development of the emergent knowledge which is the purpose of transactive resource markets.

Table 10-3 EiSubscriptionResponse Attributes

Attribute	Attribute Type	FIX Field	Meaning
MultiCast Listen Reference	String	PrimaryServiceLocationID(2567)	If present and non-null the Subscription Manager provisions the subscription by sending a reference to (e.g.) a multicast in lieu of response messages. Optional. If included, the subscription is sent via a mechanism that does not use individual responses.

Deleted: direct

Attribute	Attribute Type	FIX Field	Meaning
Response	EiResponse Type	Not in FIX	A standard CTS response type; see Section 2.4
Subscription Action Taken	Subscription Action Type (enumeration)	SubscriptionRequestType (263)	The action taken on the referenced or newly created Subscription.
Subscription Request ID	Ref ID Type	MDReqID(262)	A UID indicating the newly created, modified, or canceled subscription.

Moved (insertion) [7]

Deleted: , “ ”.

11 Tickers

This section applies the subscription pattern of Section 10, and describes mechanisms to access continuous Market Data on the activities of market participants. CTS calls these *Tickers*. Tickers update continuously, on a schedule determined by the provider, as Parties interact with a Segment.

A Party wishing to trade in a market naturally wants to be kept apprised of changing information about the market. The FIX Protocol divides this information into three categories: Orders, Trades, and Bids/Offers. CTS defines Tickers for Tenders [Orders], Transactions [Trades], Quotes [Bids/Offers], and Requests for Quotation [RFQs].

The four types of Tickers are represented as an enumeration. See Table 11-1 and Figure 11-1 below.

Table 11-1: Types of Tickers in CTS Facet

Ticker Type	Request Payload
Quotes	Published Indicative (non-Tradeable) Quotes
RFQs	Published RFQs
Tenders	Anonymized Tenders offering to Buy or to Sell
Transactions	Anonymized Trades, whether from market matches or from Negotiation

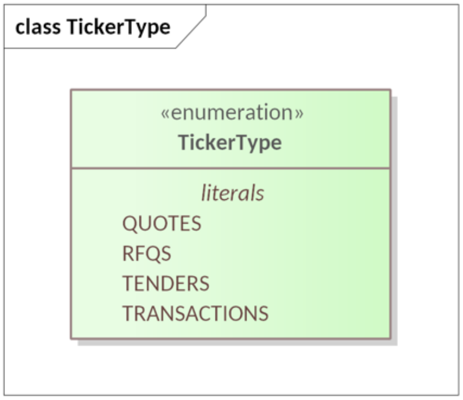


Figure 11-1 TickerType Enumeration

Not all Markets or Market Segments support Ticker subscriptions or all Ticker types. Actors can discover what Tickers a Segment supports and how to interact with them through the Market Reports as discussed in Section 13, *"Market Structure Reference* Data: Market, Segment, and Session Subscriptions*"*.

Private Quotes do not appear in Tickers.

It is common that, following market or segment rules, most parties in tickers are anonymized, that is, the identity of the party is not disclosed. In such situations, the Market Party ID is used as the Party ID and/or Counterparty ID in the Ticker.

In Resource markets as in financial markets, Parties with specific and/or influential roles are not anonymized. For example, a Market may choose not to anonymize the Party ID of the distribution system operator (DSO).

This specification makes no statement about what anonymization rules a resource market must use. This specification offers general guidance that most participants be anonymized to preserve privacy, but that Ticker messages for significant participants may be distributed under their own identity.

11.1 Messages for Tickers

An Actor subscribes to a Ticker based on the subscription model (Section 10, “Subscription Facet”). An Actor can subscribe to a single Market Segment or any or all Market Segments in a Market. Each Ticker Type, if available, requires a separate Subscription.

Table 11-2 Ticker Facet Messages

Facet	Request Payload	Response Payload	Notes
Ticker	EiManage Ticker Subscription Payload	EiManaged Ticker Subscription Payload	As multiple Markets may use same Ticker service, must allow multiple subscriptions.

11.2 Interaction Pattern for Tickers

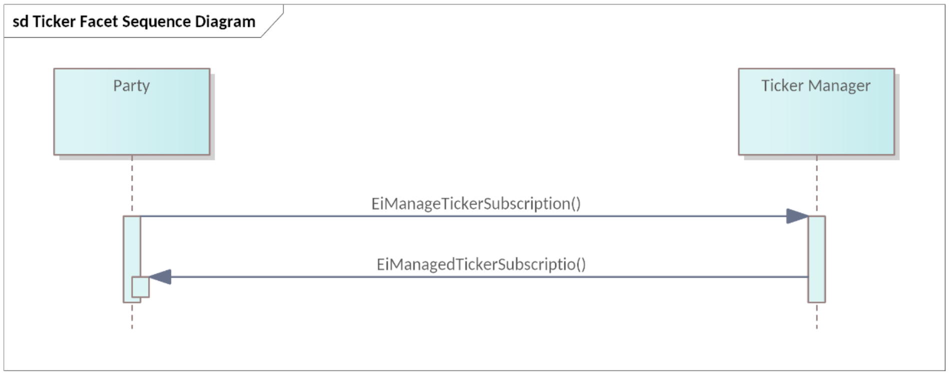

sd Ticker Facet Sequence Diagram
Party
Ticker Manager
EiManageTickerSubscription()
EiManagedTickerSubscriptio()

Figure 11-2: UML Sequence Diagram for the Ticker Facet

11.3 Exceptions to Ticker Subscription Interactions

A given Segment may provide a single Ticker data stream combining any or all Ticker Payloads. An Actor that subscribes to any Ticker implicitly subscribes to all the Types included with that Ticker.

In larger markets, there may be a broadcast or multicast channel for a Ticker. In such markets, there is no subscription; the Actor simply listens to that broadcast channel. The Subscription Id is not part of the multicast and an Actor unsubscribes as per the transport used.

11.4 Interaction Patterns for Ticker Data

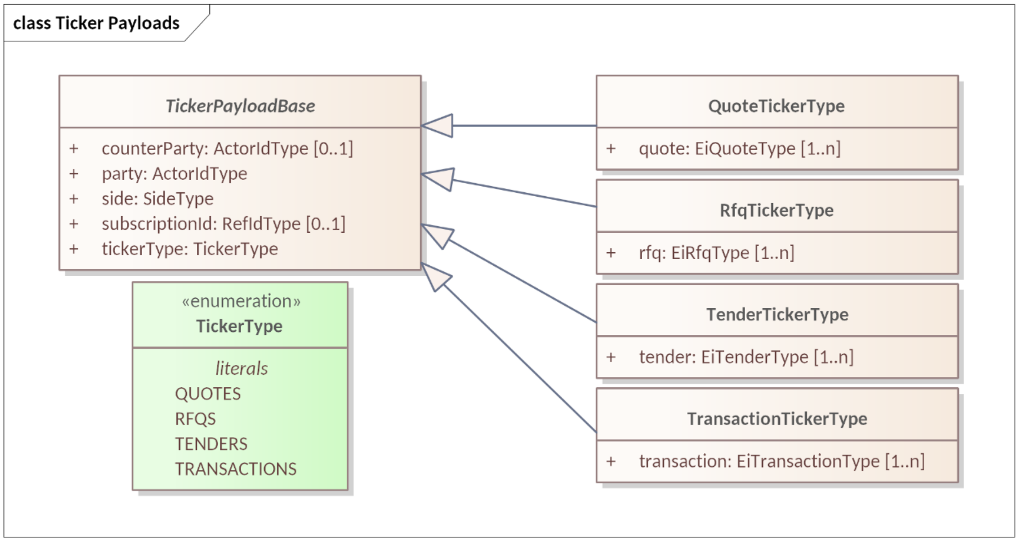
The various types of tickers share a common approach:

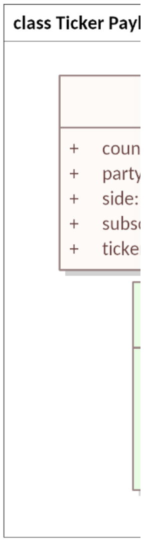
- A subscription is created using EiManageTickerSubscription, passing the requested change and which ticker is being managed.
- The ticker payloads contain the subscription ID and the relevant object for the ticker type:
    - TenderTickerType is EiTenderType for Bid and Offer tickers.
    - TransactionTickerType is EiTransactionType
    - QuoteTickerType is EiQuoteType (not tradeable)
    - RfqTickerType is EiRfqType.
- The Ticker Payloads are described below in Figure 11-3. Delivery of Ticker Payloads is out of scope,

- Large or complex markets might use a multicast for delivery using the relevant ticker payloads (out of scope)
- Small or less complex markets might use a market-defined delivery mechanism (out of scope)

11.5 Information Model for Ticker Payloads

Ticker payloads are sent asynchronously when subscribed. The UML Class Diagrams for Ticker Payloads and Ticker Type are in Figure 11-3.





Deleted:

Figure 11-3 Ticker Payloads and Ticker Type showing inheritance

The attributes for the Ticker Payloads are shown in Table 11-3. Ticker Payloads will be delivered pursuant to ticker subscriptions on a Segment.[20]

Table 11-3 Attributes for the Ticker Payload Base and Ticker Types

Attribute	Attribute Type	FIX Field	Meaning
Counter Party	Actor ID Type	PartyID(448)	The counterparty in the ticker payload by type; may be anonymized per market rules. Optional. CounterParty for Ticker Payload Base may be anonymized for Transaction, and may be included for market-facilitated Tender and RFQ tickers.
Party	Actor ID Type	PartyID(448)	The party in the referenced ticker; may be anonymized per market rules.

[20] Just as for message payloads, how these are delivered is out of scope. Some Markets and Segments may use multicast or delivery as response message(s).

Deleted: direct

Deleted: .

Attribute	Attribute Type	FIX Field	Meaning
Side	Side Type	Side(54)	The side for the referenced ticker; note that an EiTender, etc., have side in the inherited TenderBase. The Side in the Ticker Payload Base MUST match that in any referenced object.
Subscription Request ID	Ref ID Type	MDReqID(262)	An optional UID indicating the related subscription. Present only for individual subscriptions but MAY be absent even then. NOTE that if delivered via (e.g.) multicast or broadcast, customization of Subscription IDs cannot be done, so this attribute MAY be absent. Cancelation of a multicast Snapshot-and-Updates subscriptions is accomplished by sending an EiManageTicker Payload with the original Subscription Request ID.
Ticker Type	Ticker Type enumeration	Not in FIX	See Figure 11-3 for class diagram. The values are QUOTES, RFQS, TENDERS, and TRANSACTIONS
Quote	EiQuoteType		For QuoteTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
RFQ	EiRfqType		For RfqTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
Tender	EiTenderType		For TenderTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
Transaction	EiTransactionType		For TransactionTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.

Moved (insertion) [8]

11.5.1 Tender Tickers

Bids and Offers are simply Buy or Sell side Tenders. When a Tender is submitted, the Segment announces the Tender on the Ticker subject to the Segment rules and requests for publication and privacy.

Tenders are submitted to the entire Market Segment; there is no guarantee that a Tender will still be available when a Party submits a matching Tender.

Deleted: market segment

The Market and/or Segment may publish Quotes subject to Issuer's request for publication, subordinate to Market and/or Segment rules.

Deleted: Publish

Deleted: Issuer

Deleted: Publication

The payload for Tender Tickers includes one or more EiTenderType objects with attributes anonymized following market or segment rules. Attributes are shown in Table 5-2: EiTender Attributes.

A Party that wishes to receive Tenders from a Segment must subscribe to that Segment's Tender Ticker.

11.5.2 Quote Tickers

If a Segment and its Market Mechanism supports Negotiations, then it supports a Quotes Ticker. There is more diversity in Quotes than in Tenders.

The Quote attribute of the Quotes Ticker Type is defined in Section 9 “Negotiations.” Because the purpose of a public offer (“publishing a Quote”) is to initiate a Negotiation between Parties, the Quotes Ticker is not anonymized.

11.5.3 RFQ Tickers

While the type and semantics of RFQs and Quotes are closely related, the separation simplifies the data model. There may be reasons for a Negotiation market to not support an RFQ Ticker.

Deleted: negotiation

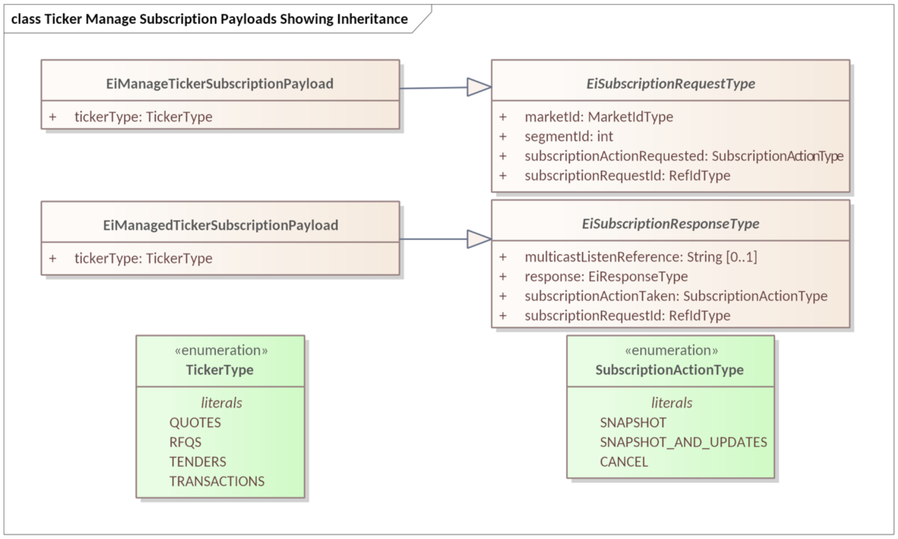
11.5.4 Transaction Tickers

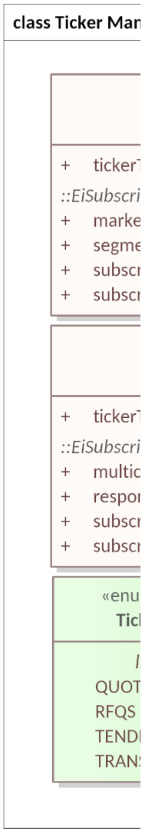
The Transactions Ticker is the continuous advertisement of Trades executed in a Market Segment. Both Parties are listed on a Transaction, although either or both may be anonymized as specified in market rules.

In some Market Mechanisms (see 13.1, *“Market Mechanisms”*) the contract may be negotiated privately. Note: even a Transaction that was negotiated privately will be published in the Transaction ticker based on market rules.

11.6 Message Payloads for Managing Ticker Subscriptions

The messages for adding, changing, or deleting a Ticker subscription contain only the ticker type and a subscription request or response as defined in Table 10-2 and Table 10-3. The UML Class Diagrams for the message payloads are shown in Figure 11-4.


class Ticker Manage Subscription Payloads Showing Inheritance
EiManageTickerSubscriptionPayload
+ tickerType: TickerType
EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiManagedTickerSubscriptionPayload
+ tickerType: TickerType
EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
TickerType
literals
QUOTES
RFQS
TENDERS
TRANSACTIONS
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL



Deleted:

Figure 11-4 Ticker Manage Subscription Payloads showing inherited attributes

Table 11-4 shows the attributes for the EiManage and EiManaged Ticker Subscription Payloads.

Table 11-4 Attributes for the EiManage and EiManagedTickerSubscription Payloadfs

Attribute	Attribute Type	FIX Field	Meaning
Ticker Type	Ticker Type enumeration		The Type of Ticker for subscription.
All other attributes are as in EiSubscriptionRequest and EiSubscriptionResponse in Table 10-2 and Table 10-3			

12 Instrument Data Subscriptions

Instrument Summaries are obtained by Subscription (described in Section 10) and provide dynamic data about specific Instruments traded in the Segment. Like other Subscriptions, Instrument Summary Subscriptions provide an aspect of what FIX calls Pre-Trade Data.

The information in the Instrument data may be considered a blend of Reference and Dynamic—the Reference Data includes all the attributes of Instrument Session Report Type except for the Instrument Summary; the Reference data in effect describes the market, segment, resource, and similar reference data. The combination is dynamic, with static identifying information.

Deleted: is both

As resource market instruments are time-based, the tradeable set of instruments may change over time as old instruments expire and become iirrelevant (perhaps post-reconciliation) and new instruments become available. Hence instrument data is always dynamic.

The request for a subscription includes the usual requests for snapshot, snapshot and updates, and unsubscribe (see Section 10) with the addition for Instrument data of how to update.

Deleted: Reference

The Subscription Manager may restrict the frequency and the content. Certain requests, e.g., multiple levels of the order book, or many instruments, involve a lot of data. Some restrictions are described in Section 13.3.3 “Information Model for Segment Reference Data” (Max Summary Instruments and Market Depth.

12.1 Messages for Instrument Data Subscriptions

Deleted: Reference

Subscription requests need additional information beyond the EiManageSubscription payloads:

- The Bounding Interval for instruments requested
- A limit on how many instruments to supply data for
- How to update the requested data if the subscription requests updates—incremental or a full update

Deleted: reference

Deleted: reference

Table 12-1 Messages for Instrument Data

Deleted: *Reference*

Facet	Request Payload	Response Payload	Notes
Subscription – Instrument Data	EiManage Instrument Data	EiManaged Instrument Data	Create, manage, and cancel subscriptions

Deleted: Ref

Deleted: Ref

Deleted: Reference

12.2 Interaction Pattern for Instrument Data Subscriptions

Deleted: Reference

An Instrument Data Subscription requests data on contiguous temporal range of Instruments.

Deleted: Reference

Within a Market Segment, trading is for a single Product, and Instruments are distinguished by the resource delivery Interval. The Subscription returns data for all Instruments whose interval falls within the Bounding Interval of the Subscription request.

Deleted: market

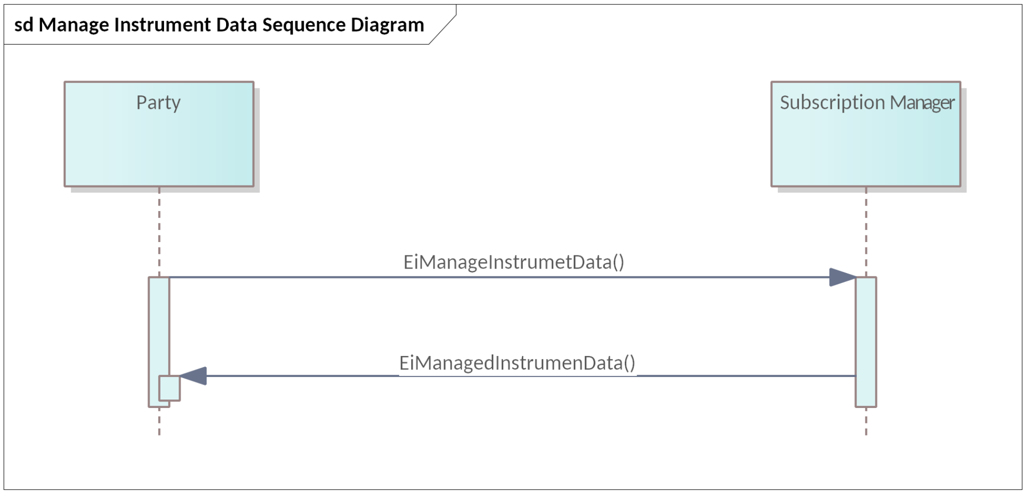


Figure 12-1 Manage Instrument Data Subscriiption

12.3 Information Model for Manage Instrument Data Subscription Payloads

The UML class diagram for the Manage Instrument Data messages is in Figure 12-2.

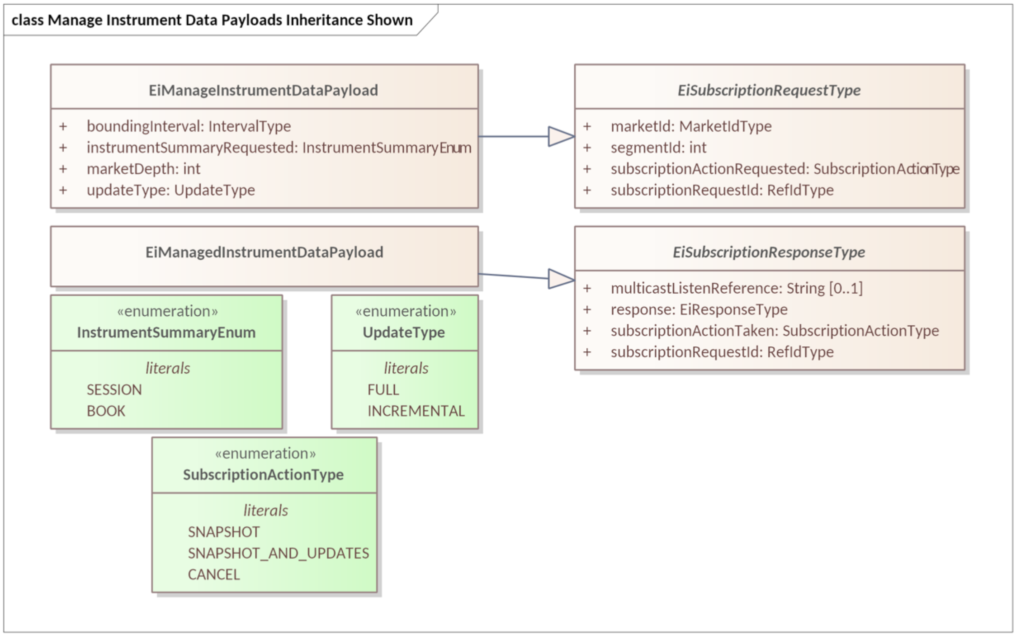
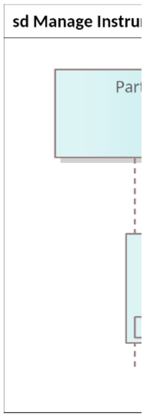

class Manage Instrument Data Payloads Inheritance Shown
EiManageInstrumentDataPayload
+ boundingInterval: IntervalType
+ instrumentSummaryRequested: InstrumentSummaryEnum
+ marketDepth: int
+ updateType: UpdateType
EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiManagedInstrumentDataPayload
EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
InstrumentSummaryEnum
literals
SESSION
BOOK
«enumeration»
UpdateType
literals
FULL
INCREMENTAL
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL

Figure 12-2 UML Class Diagram for Manage Instrument Data Messages



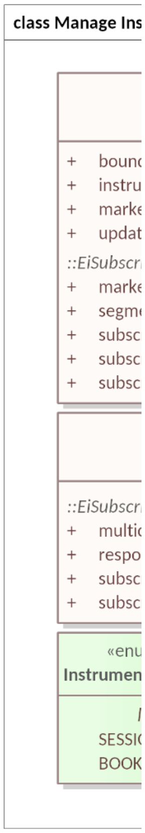
Deleted:

Deleted: *Reference*

Deleted: *Subsription*

Deleted: **Reference**

Deleted: Reference



Deleted:

Deleted: *Reference*

The Manage Instrument Data payload specifies the type of summary and instruments requested. Its attributes are in Table 12-2.

Deleted: Reference

Table 12-2: Attributes for Manage Instrument Data Payload

Attribute	Attribute Type	FIX Field	Notes
Bounding Interval	Interval Type	Not in FIX	Subscription request is for all Instruments within the Bounding Interval. What is returned is at the discretion of the Segment. The request will return information on all instruments within the [closed] time interval whose start is at or later than the Bounding Interval start and whose end point is at or before the end of the Bounding Interval. See Section 3.3 “The Bounding Interval Pattern in CTS”
Instrument Summary Type	Instrument Summary Type Enumeration	CTS	Supported values are: 0 = Session Summary (CTS: **SESSION**) 1 = Book (see Market Depth attribute) (CTS: **BOOK**)
Market Depth	Integer	MarketDepth (264)	Depth of market requested for Book Snapshot and/or Incremental updates 0 = full book depth 1 = top of book 2 or greater = book depth (number of levels). The Segment may limit the response to the depth indicated by Market Depth attribute of Segment Reference Data.
Update Type	Update Type Enumeration	MDUpdateType (265)	Enumeration. FIX values are 0 = **FULL** 1 = **INCREMENTAL** The nature and frequency of Incremental Updates is at the discretion of the Subscription Manager.
The remaining attributes are inherited from EiSubscriptionRequestType (Table 10-2)			

Moved down [10]: Not in FIX

Deleted: Type of Instrument Summary for subscription. FIX integer…

Deleted: Fix values and

The attributes for the *Managed Instrument Data Payload* are all inherited from EiSubscriptionResponseType (Table 10-3).

Deleted: *Reference*

12.4 The Instrument Session Reports

As with Tickers (See Section 11 “Tickers”) the actual requested information may be delivered by various means, including but not limited to multicast, point-to-point delivery, and publication to be downloaded by the actor, some of which may not support subscription request identifiers.

The Instrument Session Reports provide summary information about one or more instruments. Common information about the report is presented in class Instrument Report.

Deleted: (FIX calls this the Market Data Instrument Header)….Information Information about each instrument is included as attributes in what FIX calls the Market Data Instrument Summary.¶

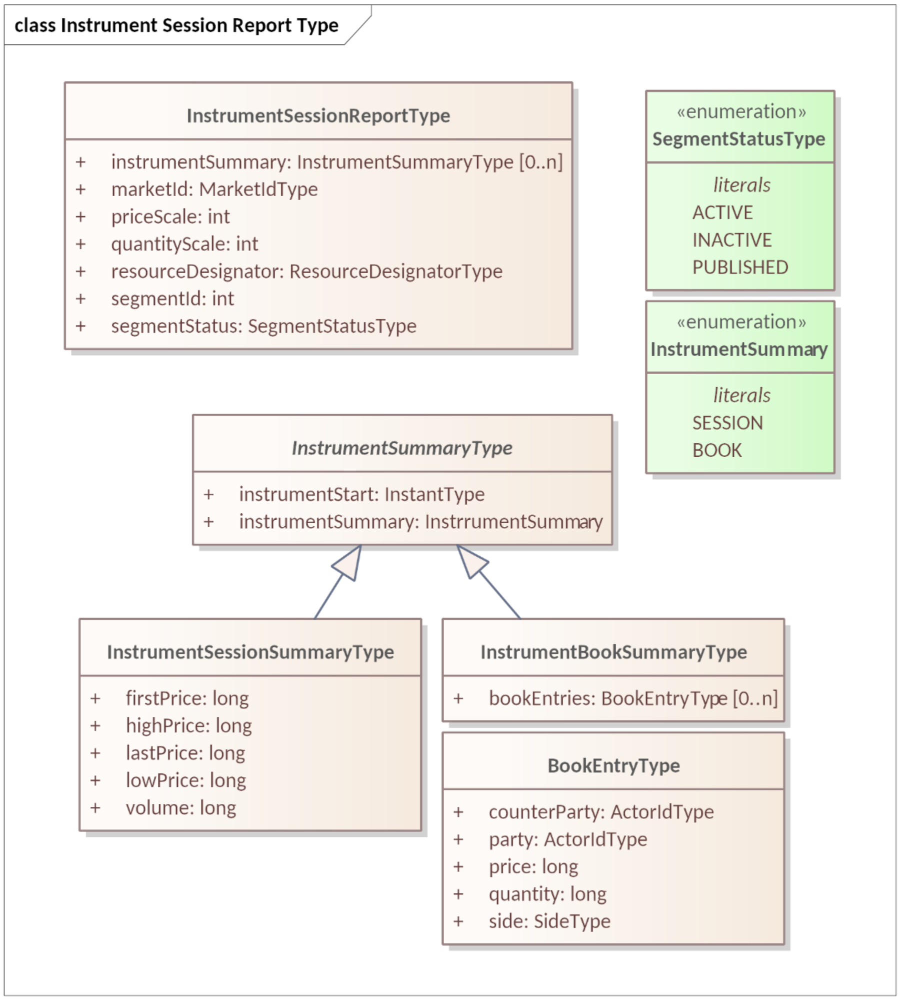
In CTS, the messages are modeled by Instrument Session Report Type which has zero or more Instrument Summary Type instances.

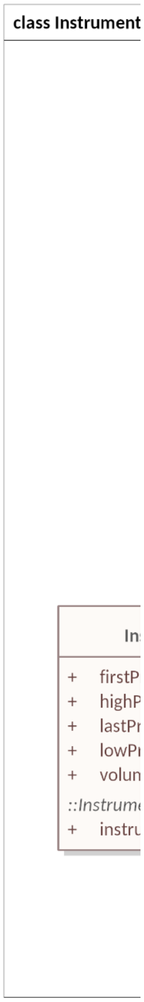
Deleted: Information about instrument is included as attributes in what FIX calls the Market Data Instrument Summary.¶

Deleted: modelled

12.4.1 Information Model for the Instrument Session Report Type

The UML class model for Instrument Session Report Type is shown in Figure 12-2.


class Instrument Session Report Type
InstrumentSessionReportType
+ instrumentSummary: InstrumentSummaryType [0..n]
+ marketId: MarketIdType
+ priceScale: int
+ quantityScale: int
+ resourceDesignator: ResourceDesignatorType
+ segmentId: int
+ segmentStatus: SegmentStatusType
«enumeration»
SegmentStatusType
literals
ACTIVE
INACTIVE
PUBLISHED
«enumeration»
InstrumentSummary
literals
SESSION
BOOK
InstrumentSummaryType
+ instrumentStart: InstantType
+ instrumentSummary: InstrrumentSummary
InstrumentSessionSummaryType
+ firstPrice: long
+ highPrice: long
+ lastPrice: long
+ lowPrice: long
+ volume: long
InstrumentBookSummaryType
+ bookEntries: BookEntryType [0..n]
BookEntryType
+ counterParty: ActorIdType
+ party: ActorIdType
+ price: long
+ quantity: long
+ side: SideType



Deleted:

Figure 12-3 UML Class Diagram for Instrument Session Report Type

The attributes for the classes in Figure 12-3 are shown below.

Table 12-3: Attributes for the Instrument Session Report Type

Attribute	Attribute Type	FIX Field	Meaning	Notes
Instrument Summary	Instrument Summary Type	Not in FIX	A repeating series for each Instrument in the Report. The information varies by the Summary Type requested.	Zero or more Instrument Summaries; type is in response to that requested in the Instrument Summary Type included in the Manage Instrument Reference Data payload,

Moved (insertion) [2]

Deleted: Series

Deleted: In FIX a

Attribute	Attribute Type	FIX Field	Meaning	Notes
Market ID	Market Id Type	MarketID (1301)	Identifies the Market	Identifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13.
Price Scale	Integer	Not in FIX	A multiplier for the Price used in this segment	A market segment might be denominated in e.g. dollars or 10ths of a cent.
Quantity Scale	Integer	UnitOfMeasure (996) UnitOfMeasureQuantity (1147)	A scale factor for the Resource unit for the Segment being reported on.	A scale factor for Resource Units; the number of resource units in a trade of quantity one of an instrument.. See Table 3-2: Defining the Product
Resource Designator	Resource Designator Enumeration	FIX Instrument Component	Identifier of the Resource being offered	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions.
Segment ID	Integer	MarketSegmentID (1300)	Unique Identifier for Segment	FIX Segment is a string to allow a UID. CTS Segment is an integer intended to be used with a MarketID UID.
Segment Status	Segment Status Type Enumeration	MarketSegStat (2542)	Segment status as of time of report	1 = Active (CTS: **ACTIVE**): Market segment is active, i.e. trading is possible. 2 = Inactive (CTS: **INACTIVE**): Market segment has previously been active and is now inactive. 3 = Published (CTS: **PUBLISHED**): Market segment information is provided prior to its first activation.

Deleted: defined

Deleted: Not defined in FIX

Deleted: multiplicative

Deleted: , e.g. 100, to convert from market

Deleted: this

Deleted: units to the base unit size traded in this Segment.…

Deleted: Segment

12.4.2 The Instrument Summary Types

The Instrument Summary is the information in an Instrument Summary Report that is repeated for each Instrument in the range.

The information conveyed varies with the Instrument Subscription Type. The UML class model for Instrument Summary Type is shown in Figure 12-4. The attributes are shown in Table 12-4, Table 12-5, and Table 12-6.

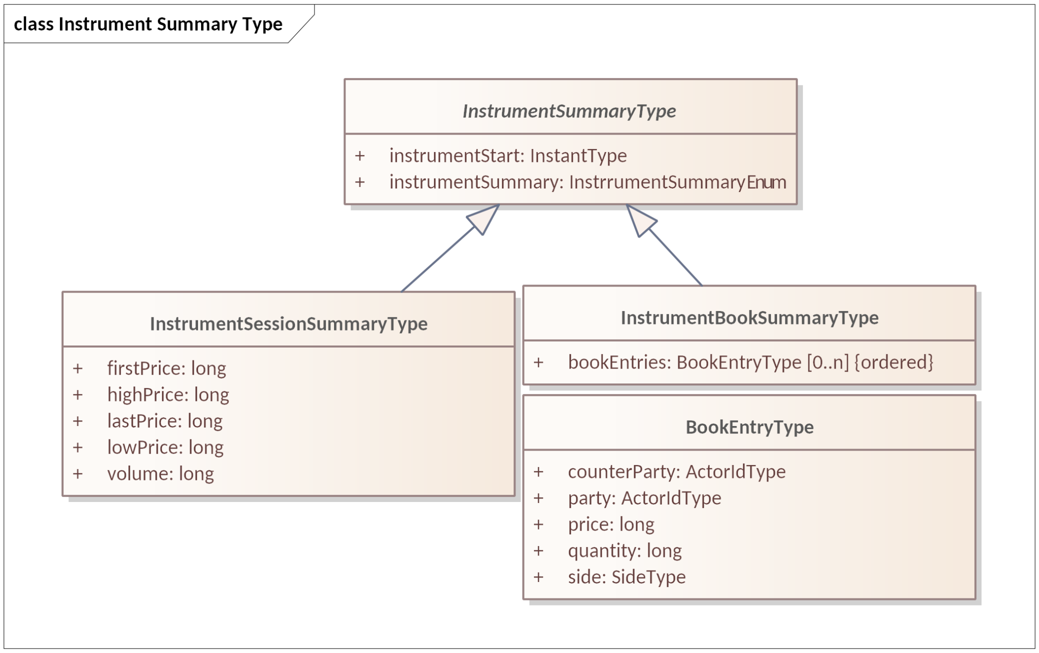

class Instrument Summary Type
InstrumentSummaryType
+ instrumentStart: InstantType
+ instrumentSummary: InstrrumentSummaryEnum
InstrumentSessionSummaryType
+ firstPrice: long
+ highPrice: long
+ lastPrice: long
+ lowPrice: long
+ volume: long
InstrumentBookSummaryType
+ bookEntries: BookEntryType [0..n] {ordered}
BookEntryType
+ counterParty: ActorIdType
+ party: ActorIdType
+ price: long
+ quantity: long
+ side: SideType

Figure 12-4 Instrument Summary Type UML Class Diagram

12.4.2.1 The Instrument Session Summary

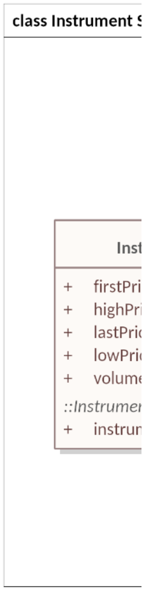
A common change reported in a Session Summary shows after a Session changes its state. In transactive resources, each Instrument closes on its own schedule. A Segment might not permit trading in an Instrument more than forty-eight hours in the future. A Segment might not permit trading an Instrument with a Start DateTime in the past. We term the union of Segment schedule and Instrument tradability the Instrument Session.

Instrument Session summaries include opening prices, closing prices, and volume traded. Note that all prices are scaled using Price Scale in the Session Report. Volume is not scaled.

The UML class diagram is in Figure 12-4

Table 12-4: Instrument Session Summary Type attributes

Attribute	Attribute Type	FIX Field		Meaning
Instrument Start	Instant Type	Not in FIX		Start time that identifies this instrument and thereby this Instrument Session Summary Detail
Instrument Summary	Instrument Summary Enum	Not in FIX		An anumeratiion indicating whether a book (order book) or session summary is provided. The information is the same as expressed in the type system.
First Price	Integer	FirstPx (1025)		Indicates the first price of a Session; can be a bid, offer, or trade price.
High Price	Integer	HighPx (332)		The high end of the price range prior to the open or reopen



Deleted:

Deleted: announces when

Deleted: FIX Attribute Type

Deleted Cells

Deleted: DATETIME

Deleted: PRICE

Deleted Cells

Deleted: PRICE

Attribute	Attribute Type	FIX Field		Meaning
Last Price	Integer	LastPx (31)		Indicates the last price of a Session; can be a bid, offer, or trade price.
Low Price	Integer	LowPx (333)		The low end of the price range prior to the open or reopen
Volume	Integer	TotalVolumeTraded(387)		Total volume traded of an instrument, including negotiated and market trades.

Deleted: FIX Attribute Type

Deleted Cells

Deleted: PRICE

Deleted: PRICE

Deleted: Total Volume Traded …

Deleted: QUANTITY

12.4.2.2 The Instrument Book Summary

The Book is the set of all Tenders, including Tradeable Quotes, in the Market Segment. In an active market, unless there are restrictions on matching, all Tenders to sell are priced higher than all Tenders to buy; if there were an overlap, they would already have generated Transactions and the Tenders would be removed.

The depth of the Book is a list of the volume bid or offered at each price. The Book sorts Bids by descending price. The Book sorts Offers by ascending price. A Top of the Book request, subscription depth of 1, provides just the top entry in each list, anonymized. A subscription depth of 0 provides both entire sorted lists, anonymized. Any other subscription level (n) provides the first (n) entries in each level.

The UML class diagram is in Figure 12-4.

Table 12-5: Instrument Book Summary Attributes Including those Inherited from Instrument Summary Type

Attribute	Attribute Type	FIX Field	Meaning
Instrument Start	Instant Type	Not in FIX	Time stamp (inherited from Instrument Summary Type)
Instrument Summary	Instrument Summary Enum	Not in FIX	An anumeratiion indiicating whether a book (order book) or session summary is provided. The information is the same as expressed in the type system.
Book Entries	Book Entry Type	NoMDEntries(269)	An ordered repeating element for each side and level of the Book

Moved (insertion) [3]

Deleted: Begin DateTime

Deleted: Repeating

The Book Entry is the repeating information for each Side in the Book. The Book Entry is the same message format as a Quote, anonymized as required by market rules. Book Entry attributes are in Table 12-6. The UML class diagram is in Figure 12-4.

Table 12-6 Book Entry Attributes

Attribute	Attribute Type	FIX Field		Meaning
Party	Actor ID Type	PartyID (448)		Party for the specific Side Type. MAY be anonymized following Market Rules.
CounterParty	Actor ID Type	PartyID (448)		CounterParty for the specific Side Type. MAY be anonymized following Market Rules.
Price	Long	MDEntryPx(270)		Price in the book. Subject to Price Scale.

Deleted: FIX Attribute Type

Deleted Cells

Deleted: Counter

Deleted: 488

Deleted: String

Deleted: Party

Deleted: 488

Deleted: String

Deleted: Party

Deleted: Price

Attribute	Attribute Type	FIX Field		Meaning
Quantity	Long	MDEntrySize(271)		Quantity in the book. Subject to Quantity Scale.
Side	Side Type	MDEntryType(269)		On which side is the Entry

Deleted: FIX Attribute Type

Deleted Cells

Deleted: Qty

Deleted: Side (54)

Deleted: Char

Deleted: Party?

13 Market Structure Reference Data: Market, Segment, and Session Subscriptions

For any Market, there are standing terms and expectations about Product offerings. If these standing terms and expectations are not known, a Party may have to use many interactions to discover where to trade for the Products that meet that Party’s needs.

For the Trader, the questions include

- “What products are traded in this Market, and where are they traded?” (Market Structure)
- “How and when can I trade in each Segment in this market? (Segment description)
- “What instruments can I trade now (Session information)

Deleted: venue

Deleted: “When can I trade and what…

CTS uses the standard mechanism of the CTS Subscription to query the Market structure including enumerating the Segments, to describe each Segment, and the status of the current trading session in each Segment. A Trading Session is a period for trading in a Segment between the opening and the closing of the Segment.

In CTS Markets, the Instruments tradeable in a Trading Session may change regularly as Instruments enter or exit the trading window of the Market.

A Party must interact with a specific Segment to trade a specific Product. A Market MAY contain two or more Market Segments trading the same Product; such segments may differ in the Market Mechanism, or in trading window. For example, a regulated provider may offer a day-ahead hourly market based on an Auction between 9:00 AM and 3:00 PM. The same actor may trade the same Product by order book in another Segment. The Auction and the Order Book are different mechanisms for matching buyer and seller.

A Party chooses to trade in the Segment that it anticipates will be to its greatest advantage. The Party will make this choice based on anticipated price, or on block size, or even on Warrants. Because Transactions are committed when created, a Party may buy on one Segment, and thereafter sell part of it on another. Segments may be available for trading on different schedules, and the Instruments available in each Segment change over time. The Segment Structure provides detailed information to guide trading, negotiation, and settlement. The Segment Structure defines when Sessions open and when Sessions close.

All trades occur within a Trading Session. Trading Session Data provides information on trading in a Session, including times for session changes (open, close, and more) and tradeable instruments . Trading Session Status informs whether a Session is available for trading, and when that status will change. A Trading Session’s Tradeable Instrument Trading Range permints a Party to compute whether an Instrument is currently tradeable.

Deleted: in

Deleted: Sessions

Deleted: over time

Deleted: Segment

Deleted: enables

Information on instruments is described iin Section 12.4.

A Party discovers a Market, including changes over time, by subscribing to Market Structure Reference Data. Market Structure Reference Data includes a description of all Segments in the Market. A Party discovers and monitors a Segment by subscribing to the Segment Reference Data. A Party monitors the changing constraints on a Segment by subscribing to Trading Session Data.

This Section describes the interactions to subscribe to Market Reference Data, to Segment Reference Data, and to Trading Session Data.

13.1 Market Mechanisms

One of the most important distinctions between Segments is the Market Mechanism. The FIX Trading Community defines a standard called Market Model Typology **[MMT]**. MMT classifies the mechanisms and general algorithms that operate a Market.

Deleted: Types

A Party participating in trading may change its behavior based on the mechanism the Segment uses to execute trades. The optimum trading strategy for a Party will change between an order book and an auction. If there is only a single seller, the Buyer will want to attend closely to the quotes from that seller.

Deleted: settle

Deleted: buyer

CTS characterizes each Segment in part by its mechanism. FIX MMT defines some mechanisms that are not included by CTS. CTS also supports mechanisms not included in FIX, such as a self-executing mechanism to settle the difference between consumption as measured at the Meter (Delivery) and the Position as known (see Section 7. "The Position Facet" and Section 8 "The Delivery Facet").

Deleted: The

Figure 13-1 shows the UML Class Diagram for the Market Mechanism Enumeration. Detailed description is in Table 13-1 showing the Market Mechanism Types supported by CTS and the FIX MMT information.

A Market Mechanism is an attribute of a Segment; Sessions related to each Segment share the ordinary trading mode from the Segment MMT, but a Session MAY have a different trading mode, e.g. "scheduled opening auction" or "unscheduled auction", which is described in Section 13.4 "Trading Session Data".

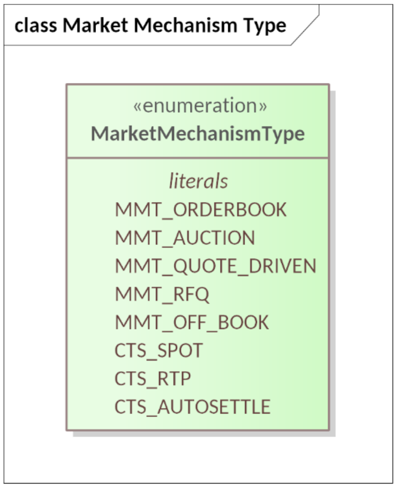


Figure 13-1 Market Mechanism Type Enumeration

Table 13-1 Market Mechanism Types in CTS

MMT Code	MMT Name	CTS Enumeration	Meaning
LB	Centralized Limit Order Book	MMT_ORDERBOOK	Participants submit their buy and sell orders, which are matched based on specific rules and executed accordingly.
PA	Periodic Auction	MMT_AUCTION	An Auction Driven Market matches Tenders only in scheduled auctions wherein all participants clear at the same price. In existing power markets, also referred to as a "Double Auction", that is, an auction in which both sellers and buyers submit bids.
QB	Quote Driven Market	MMT_QUOTE_DRIVEN	Quote Driven Markets are used for Markets with one or more dominant suppliers. Parties are notified of the Quoted price for each Instrument and submit Tenders in Quote Responses.

MMT Code	MMT Name	CTS Enumeration	Meaning
RQ	Request for Quotes	MMT_RFQ	A Request for Quotes Market is used for bilateral negotiations around price. Sellers may advertise round lots that they would like to buy or to sell, and to indicate an interest in buying or selling. Trades in a Request for Quotes Market may be for odd lots, for custom durations, and span the temporal boundaries of Products
OB	Off Book	MMT_OFF_BOOK	CTS reserves Off Book mechanisms for direct allocations of Resources from one Party to another. The Segment notifies the Parties executing the Transaction.
SM	Spot Market (CTS only; not in MMT)	CTS_SPOT	A Ticker in a Spot Market indicates the special price in the Segment that the Segment will use for "instant" purchases or sales, e.g. due to a transient or emergency situation related to the resource.
RT	Real Time Pricing (CTS only; not in MMT)	CTS_RTP	A Ticker broadcasts Indicative Quotes. Parties make no Tenders but consume a resource (as needed). Later, an Automatic Settlement Segment will generate Transactions based on Delivery.
AS	Automatic Settlement (CTS only; not in MMT)	CTS_AUTOSETTLE	Automatic Settlement creates Trades to align with consumption as measured at the meter (Delivery). Automatic Settlement self-executes Transactions for Resources consumed without previously being bought. Automated Settlement occurs in any Market in which Delivery (consumption) is not limited to prior Position.

A non-normative discussion about trading in Segments with each mechanism can be found in Appendix B Choosing a Market Mechanism.

13.2 Market Reference Data

13.2.1 Messages for Market Structure Reference Data

The payloads for Market Reference Data are shown below.

Table 13-2 Messages for Market Reference Data

Facet	CTS Initial Message	CTS Response Message	Meaning
Market Reference Data	EiManage Market Reference Data	EiManaged Market Reference Data	Request reference data for a Market

Deleted: (FIX Term is *Exchange*)….

13.2.2 Interaction Pattern for Market Reference Data

The Market Reference Data subscription enables an Actor to request the details of a Market and its Segments. The initial request returns the Market and all Segments. Update reports occur when there is a change to a Segment or to Market Reference Data, and include the Market Reference Data plus only the changed Market Segment(s). A request to cancel the Subscription suspends all further updates.

See Section 10 *“Subscription Facet.”*

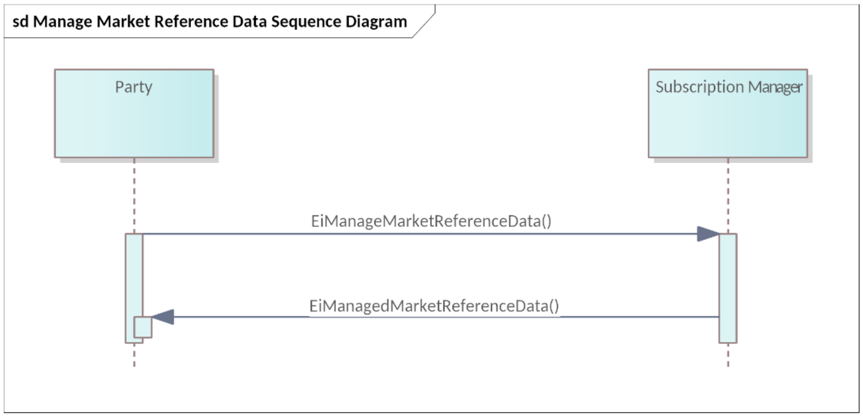
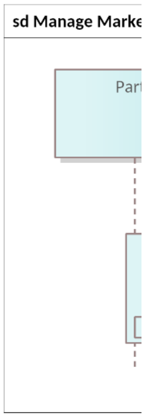

sd Manage Market Reference Data Sequence Diagram
Party
Subscription Manager
EiManageMarketReferenceData()
EiManagedMarketReferenceData()

Figure 13-2: UML Sequence Diagram for Market Reference Data

A Party may watch changes to a single Segment by naming that Segment in the subscription request. This will return only that Segment and updates to that Segment.



Deleted:

Deleted: *the*

13.2.3 Information Model for Market Reference Data

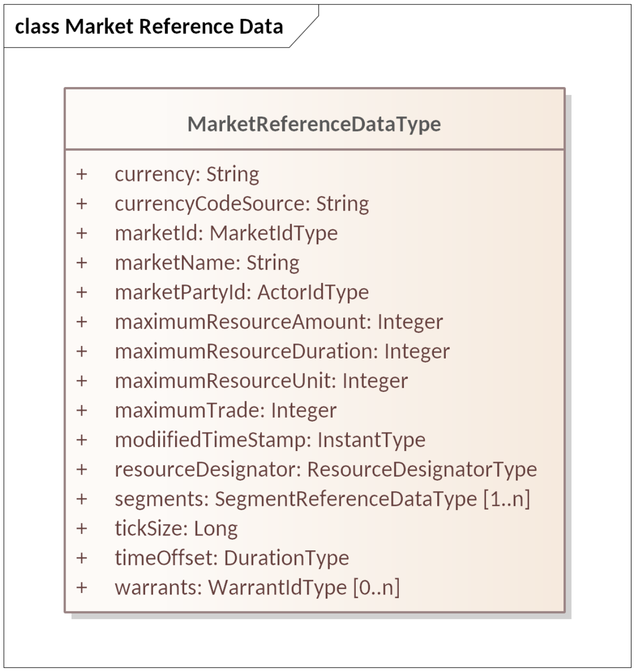
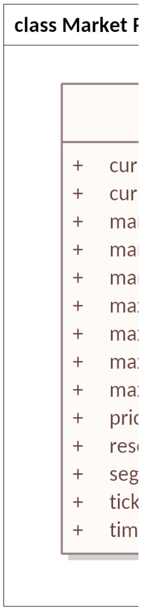

class Market Reference Data
MarketReferenceDataType
+ currency: String
+ currencyCodeSource: String
+ marketId: MarketIdType
+ marketName: String
+ marketPartyId: ActorIdType
+ maximumResourceAmount: Integer
+ maximumResourceDuration: Integer
+ maximumResourceUnit: Integer
+ maximumTrade: Integer
+ modiifiedTimeStamp: InstantType
+ resourceDesignator: ResourceDesignatorType
+ segments: SegmentReferenceDataType [1..n]
+ tickSize: Long
+ timeOffset: DurationType
+ warrants: WarrantIdType [0..n]

Figure 13-3: UML Class Diagram for Market Reference Data

Attributes for Market Reference Data are described in Table 13-3

Table 13-3 Attributes for Market Reference Data

Attribute	Attribute Type	FIX Field	Meaning
Currency	String	Currency(15)	String indicating how value is denominated in a market.
Currency Code Source	String	CurrencyCode Source(2897)	ISO – Fiat Currency per ISO 4217 DTI – Digital Token Identifier LOC – Locally defined Currency
Market ID	Market ID Type	MarketID(1301)	Note that in FIX, this is generally a formal identifier (e.g.) NYSE. If the market is a house, there is no place to look this up. There is always a UID for a Market.
Market Name	String	Not in FIX	Text providing a descriptive name for a Market. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Market Party Id	Actor ID Type	PartyID(448)	The PartyID used in Tenders to the Market and in Transactions with the Market. May also be used for anonymization of Parties.



Deleted:

Deleted: (537

Deleted: Currency Co (

Deleted: Party Id

Attribute	Attribute Type	FIX Field	Meaning
Maximum Resource Amount	Integer		Maximum Quantity of Resource Units per Maximum Resource Duration that the Market will permit.
Maximum Resource Duration	Duration		Duration for Maximum Resource Quantity
Maximum Resource Unit	Integer		Units for Maximum Resource Flow per Duration.
Maximum Trade	Integer		The value of the largest trade that the Market permits.
Modified TimeStamp	InstantType	Not in FIX	The time stamp for when the Market Reference Data was last modified.[21]
Resource Designator	Resource Designator Type		The Resource traded in this Market and Segment
Segments	Segment Reference Data Type	Market Segment	A list of one or more Market Segment descriptions for each Market Segment contained in the Market. See Section 13.3 “Segment Reference Data”
Tick Size	Integer	Tick Increment (1208)	Specifies the valid price increments at which a Party may quote or trade an Instrument.[22]. Use if a common Tick Increment required for all Market Segments. Tick Increments can increase market liquidity. Tick Size is a price and is scaled using PriceScale.
Time Offset	Duration Type		A Duration that some Markets MAY use to describe trading where a first interval is not on an “natural” boundary.[23] For example, a market in one hour Power MAY start at 7 minutes after the hour.
Warrants	Warrant ID Type		Zero or more warrants that may be suppoted in one or more segments of this market. Each segment’s reference data indicates which warrants are used. This specification does not define warrants.

Deleted: Maximum Resource Unit

Deleted: Maximum Trade

Moved (insertion) [9]

Deleted: Price Scale

Deleted: Integer

Deleted: Used to avoid floating point numbers in prices.…

Deleted: Resource

Deleted: T_OFFSET

[21] This attribute is also in Segment Reference Data; Session Data is more dynamic, so that timestamp is when the Session Data was packaged to send.

[22] Integer operations are typically much more efficient than fixed or floating point, so it is likely to be much faster to apply decimal shift on input and output rather than for more frequent comparison operations in the Execution Engine implementation.

[23] A power distribution entity may experience disruption if there is a big price change on the hour. For example, a distribution system operator (DSO) that operates multiple CTS Markets could opt to set a different offset on each Market Segment operated out of a given substation. In this model, a Market could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.

13.2.4 Payloads for Market Reference Data

The following Figure 13-4 shows the UML Class Diagram for the Market Reference Data [Subscription] payloads.

The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.

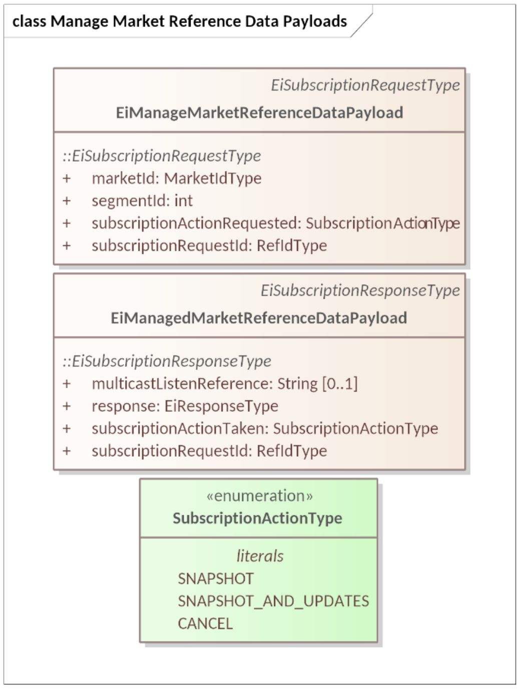
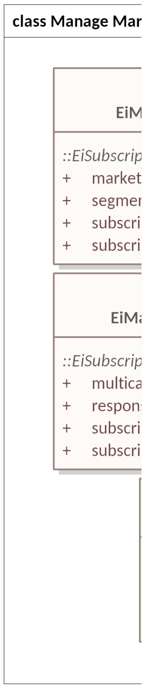

class Manage Market Reference Data Payloads
EiSubscriptionRequestType
EiManageMarketReferenceDataPayload
::EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiSubscriptionResponseType
EiManagedMarketReferenceDataPayload
::EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL

Figure 13-4 UML Class Diagram for Market Reference Data Subscription Payloads

Deleted:



13.3 Segment Reference Data

A Party must interact with a specific Trading Session to trade a specific Product. A Market MAY contain two or more Segments trading the same Product; such segments may differ in Market Mechanism, or in schedule.

A Party chooses the Segment that it anticipates will be to its greatest advantage. The Party will make this choice based on anticipated price, or on block size, or even on Warrant. Because Transactions are committed when created, a Party may buy on one Market Segment, and thereafter sell part of it on another.

A Party discovers Market Structure, including changes over time, by subscribing to that Market and/or its Segments. Even without market activity, the information provided by a Subscription may change. For example, a Segment may open or close and the biddable Instruments change regularly.

13.3.1 Messages for Segment Reference Data

Table 13-4 Messages for Segment Reference Data

Facet	Request Payload	Response Payload	Notes
Reference Data	EiManage Segment Reference Data Payload	EiManaged Segment Reference Data Payload	Messages are subclasses of the Subscription Management Messages

13.3.2 Interaction Pattern for Segment Reference Data

Figure 13-5 shows the UML Sequence Diagram for Segment Reference Data.

See Section 10 *"Subscription Facet."*

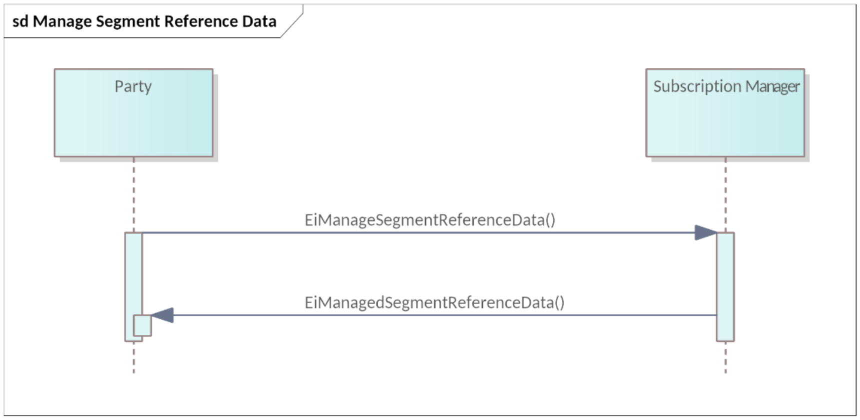

sd Manage Segment Reference Data
Party
Subscription Manager
EiManageSegmentReferenceData()
EiManagedSegmentReferenceData()

Figure 13-5 UML Sequence Diagram for Segment Reference Data

13.3.3 Information Model for Segment Reference Data

Segment Reference Data is relatively static, as Segments in typical use are long-lived.

The UML Class Diagram for Segment Reference Data is in Figure 13-6; the attribute definitions follow in Table 13-5.

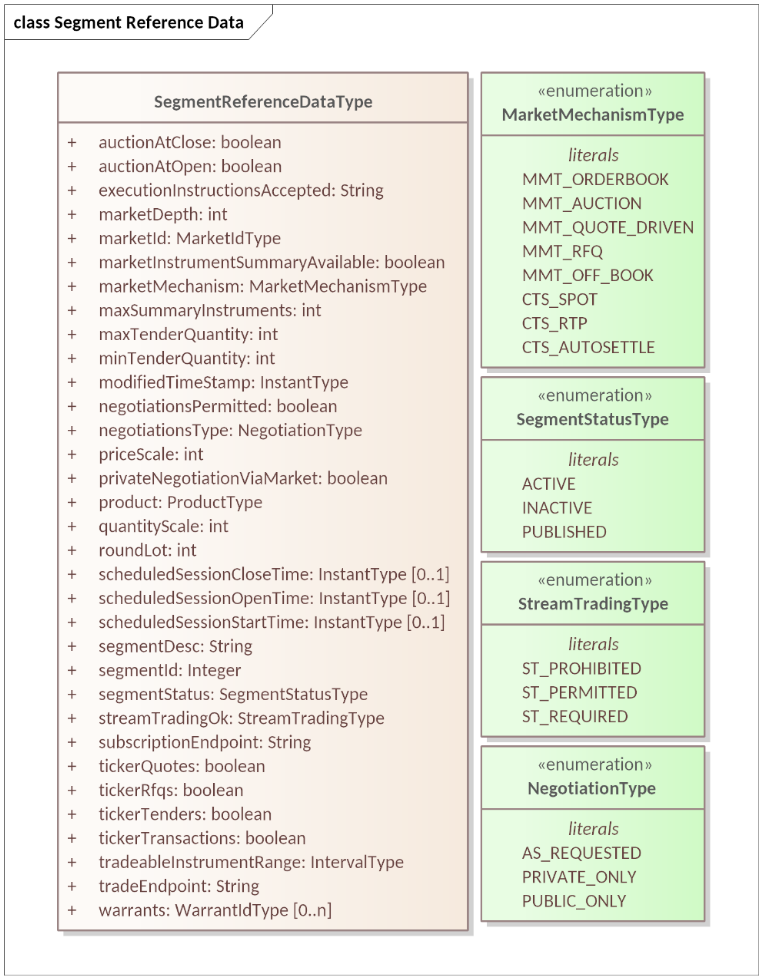
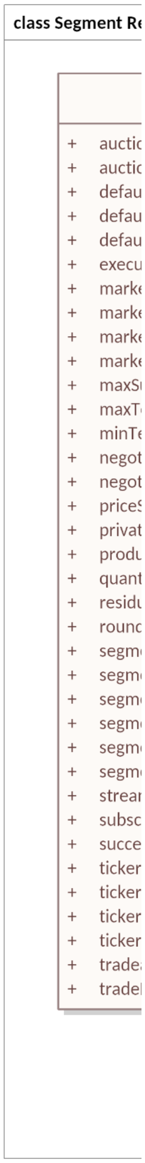

class Segment Reference Data
SegmentReferenceDataType
+ auctionAtClose: boolean
+ auctionAtOpen: boolean
+ executionInstructionsAccepted: String
+ marketDepth: int
+ marketId: MarketIdType
+ marketInstrumentSummaryAvailable: boolean
+ marketMechanism: MarketMechanismType
+ maxSummaryInstruments: int
+ maxTenderQuantity: int
+ minTenderQuantity: int
+ modifiedTimeStamp: InstantType
+ negotiationsPermitted: boolean
+ negotiationsType: NegotiationType
+ priceScale: int
+ privateNegotiationViaMarket: boolean
+ product: ProductType
+ quantityScale: int
+ roundLot: int
+ scheduledSessionCloseTime: InstantType [0..1]
+ scheduledSessionOpenTime: InstantType [0..1]
+ scheduledSessionStartTime: InstantType [0..1]
+ segmentDesc: String
+ segmentId: Integer
+ segmentStatus: SegmentStatusType
+ streamTradingOk: StreamTradingType
+ subscriptionEndpoint: String
+ tickerQuotes: boolean
+ tickerRfqs: boolean
+ tickerTenders: boolean
+ tickerTransactions: boolean
+ tradeableInstrumentRange: IntervalType
+ tradeEndpoint: String
+ warrants: WarrantIdType [0..n]
«enumeration»
MarketMechanismType
literals
MMT_ORDERBOOK
MMT_AUCTION
MMT_QUOTE_DRIVEN
MMT_RFQ
MMT_OFF_BOOK
CTS_SPOT
CTS_RTP
CTS_AUTOSETTLE
«enumeration»
SegmentStatusType
literals
ACTIVE
INACTIVE
PUBLISHED
«enumeration»
StreamTradingType
literals
ST_PROHIBITED
ST_PERMITTED
ST_REQUIRED
«enumeration»
NegotiationType
literals
AS_REQUESTED
PRIVATE_ONLY
PUBLIC_ONLY

Figure 13-6 UML Class Diagram for Segment Reference Data

Deleted:



The following table lists the attributes in the Segment Reference Data class shown above; certain attributes are present in both the Segment Reference Data and in the Session Data (See Section 13.4 “Trading Session Data” below.

Table 13-5 Segment Reference Data

Attribute	Attribute Type	FIX Field	Meaning	Comments
Auction at Close	Boolean		Scheduled behavior for related Sessions.	Related to Session second level MMT. Current session mode is in Session Data.
Auction at Open	Boolean		Scheduled behavior for related Sessions.	Related to Session second level MMT Current session mode is in Session Data.

Attribute	Attribute Type	FIX Field	Meaning	Comments
Execution Instructions Accepted	String	ExecInst(18)	A list of Execution Instructions that are accepted in this Segment (see Table 5-4).	
Market Depth	Integer	Market Depth (264)	Levels of Book that can be requested	0 – Unlimited 1-N – 0 – Unlimited 1-N – Up to N
Market ID	Market ID Type	MarketID(1301)	Identifies the containing market	
Market Instrument Summary Available	Boolean		If FALSE, no Market Instrument Summary is available	Optional
Market Mechanism	Market Mechanism Type Enumeration	MMT is not in FIX Protocol, but is from a separate activity of FIX.	Description of mechanism used to match and execute trades.	This is the default Mechanism Type during a Session for this Segment (for (continuous) trading). Sessions may use various MMT Level 2 Market Mechanisms from time to time. [24] See Section 13.1 “Market Mechanisms” and Section 13.4 “Trading Session Data”.
Max Summary Instruments	Integer		0 – U Unlimited Instruments 1-N – Maximum Instruments in a Subscription	If Market Instrument Summary Available is False, this value is ignored.
Max Tender Quantity	Integer	MaxTradeVol (1140)	The maximum order quantity in units of Instruments that can be submitted.	FIX TradeVolType (1786) allows round lots or units n (of an instrument). The default is Units; CTS uses units of the instrument. This is the maximum quantity that can be tendered or a quote lifted. Some Segments MAY set different limits for different Parties.

Deleted: FIX

Moved (insertion) [11]

Deleted: (Optional)

Deleted: Analogous to Venue Type…

Deleted: the

Deleted: the Mechanism

[24] For example, a Session might pre-open with an Auction, followed by Order Book for the bulk of the session, and end with a closing Auction.

Attribute	Attribute Type	FIX Field	Meaning	Comments
Min Tender Quantity	Integer	MinTradeVol(562)	The minimum number of units that can be ordered.	This is the minimum quantity that can be tendered or a quote lifted.
Modified TimeStamp	InstantType	Not in FIX	The time stamp for when the Session Reference Data was last modified	Also in Market Reference Data; Session Data is more dynamic, so that timestamp is when the Session Data was packaged to send.
Negotiations Permitted	Boolean	Not in FIX	Segment supports Negotiation	(Optional except Mandatory for MMT "RQ")
Negotiations Type	Negotiation Type Enumeration	(Optional except Mandatory for MMT "RQ")	Segment supports the indicated style of negotiation	Private Quotes Only (CTS: PRIVATE_ONLY) Public Quotes Only (CTS: PUBLIC_ONLY) As Requested (CTS: AS_REQUESTED)
Price Scale	Integer	Not in FIX	A multiplier for the Price used in this segment	A market segment might be denominated in e.g. dollars or cents[26] based on Market Reference Data Currency.
Private Negotiation via Market	Boolean	Not in FIX	Private Quotes are sent to the Segment which then forwards them to Counterparties	False – Prohibited – Private Quotes not forwarded by Segment True – Permitted – Segment forwards Private Quotes to listed CounterParties (FIX uses 0 to represent False, 1 to represent True)
Product	Product Type	Not in FIX	Product transactable this Segment. See Defining Product (Section 3.1.2) for details.	Each Product shares a Resource with the Market
Quantity Scale	Integer	UnitOfMeasure(996) UnitOfMeasureQty (1147)	A scale factor for the Resource Unit for this Segment	A factor to convert from market quantity units to the base unit size. See Table 3-2: Defining the Product

Deleted: MinQty (110)

Moved (insertion) [10]

Deleted: defined

Deleted: 10ths of a cent.[25]

Deleted: Not defined in FIX

Deleted: multiplicative

Deleted: , e.g. 100,

Deleted: unit

Deleted: [27]

[26] In a Segment with Price Scale of 100, a trade price of one is one one-hundredth of the intrinsic currency from Market Reference Data—price tendered is in cents if the currency is USD.

Attribute	Attribute Type	FIX Field	Meaning	Comments
Round Lot	Integer	RoundLot(561)	The trading lot size for an instrument. Chunking quantity for which a Tender may be submitted	For example, for Round Lot of 10, Tenders of 10 and 20 are accepted, and Tenders of 17 are rejected. This is an attribute of the Segment.
Scheduled Session Close Time[28]	Instant Type	TradSesCloseTime (344)	Closing Time of the trading session. Date and Time current Session next Closes (or when last session Closed)	Session times may vary for different Market Mechanisms (See Section 13.1) FIX uses UTC Time Stamps; CTS Instant Type is consistent with ISO 8601 Session Data includes the Session's actual Start, Open, and Close Times; if present in Segment Reference Data these are the default or typical session times; if session data is available the session data controls.
Scheduled Session Open Time	Instant Type	TradSesOpenTime (342)	Opening Time of the trading session. Date and Time current Session next opens (or when current or last session Opened)	Session Data includes the Session's typical Start, Open, and Close Times; if present these are the default or typical session times.
Scheduled Session Start Time	Instant Type	TradSesStartTime (341)	Starting Time of the trading session. Date and Time when Tenders may first be submitted for the current or next Session	Session Data includes the Session's typical Start, Open, and Close Times; if present these are the default or typical session times.
Segment Desc	String	MarketSegmentDesc (1396)	Text providing a description for the Market Segment. MAY be the null string.	While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.

Deleted: TradSes CloseTime…

Deleted: TradSes OpenTime

Deleted: TradSes StartTime

Moved up [11]: Optional

Deleted: Market Segment Desc…

Deleted: text

[28] Note that session data may not be available; the session close, open, and start times should match session data times if available.

Attribute	Attribute Type	FIX Field	Meaning	Comments
Segment ID	Integer	MarketSegmentID (1300)	Unique Identifier for Segment	This is a unique when considered with the Market ID.
Segment Status	Segment Status Type Enumeration	MarketSegmentStatus (2542)	Current trading status of the Market Segment.	1 = Active: (CTS: ACTIVE) Market segment is active, i.e. trading is possible. 2 = Inactive: (CTS: INACTIVE) Market segment has previously been active and is not currently Open. 3 = Published: (CTS: PUBLISHED) Market segment information is provided prior to its first activation.
Stream Trading OK	Stream Trading OK Enumeration	Stream Trading is analogous to what FIX terms multi-leg orders, in which all instruments are [bought] or none.	Applies to both Tenders and Quotes	0 – Prohibited (default if missing) (CTS: ST_PROHIBITED) 1 – Permitted (CTS: ST_PERMITTED) 2 – Required (CTS: ST_REQUIRED)
Subscription Endpoint	String		Endpoint for subscriptions to Segment	May be the same as the Trade Endpoint, Segment-specific, the same across a Market, or specific to an Actor.
Ticker Quotes	Boolean		A Ticker is available for Quotes in this Segment	True – Available for this segment
Ticker RFQs	Boolean		A Ticker is available for RFQs in this Segment.	True – Available for this segment
Ticker Tenders	Boolean		A Ticker is available for Tenders in this Segment.	True – Available for this segment
Ticker Transactions	Boolean		A Ticker is available for Transactions in this Segment.	True – Available for this segment. Transactions Ticker shows Matched Tenders and completed Negotiations in this Segment.

Deleted: SEGMENT

Deleted: Market Segment Status…

Deleted: Endpoint

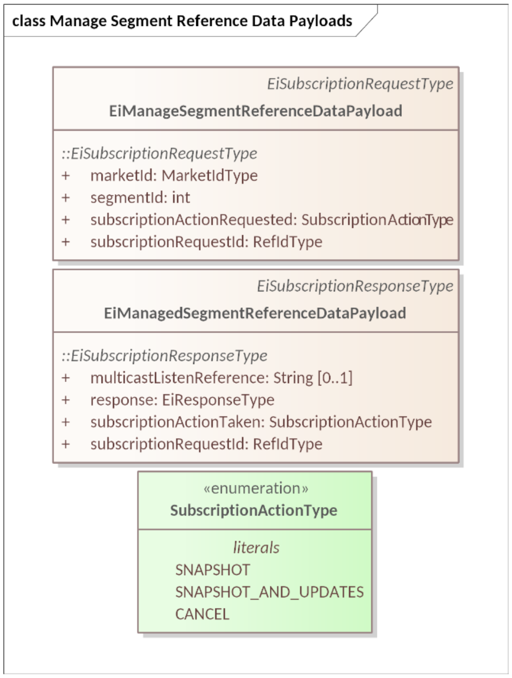
Attribute	Attribute Type	FIX Field	Meaning	Comments
Tradeable Instrument Range	Interval Type	Not in FIX	Instruments whose Interval is contained in the Tradeable Instrument Range may be traded.	Uses the Bounding Interval pattern (See Section 3.3 “The Bounding Interval Pattern in CTS”) Sessions have their own Tradeable Instrument Range which may be more dynamic.
Trade Endpoint	String		Endpoint to access trade facets of the Segment.	May be Segment-specific, the same across a Market, or specific to an Actor.
Warrants (Optional)	Warrant ID Type	Not in FIX	Optional further specificity of Product.	Warrants that MAY be available are itemized in the Market. The Segment identifies available warrants in this segment. This specification does not define Warrants.

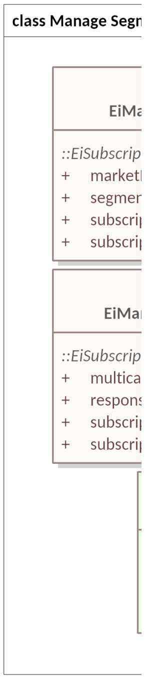
Deleted: Endpoint

13.3.4 Payloads for Segment Reference Data

The payloads for managing Segment Reference Data are subclasses of the Subscription Management Messages. See Figure 13-7.

The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.


class Manage Segment Reference Data Payloads
EiSubscriptionRequestType
EiManageSegmentReferenceDataPayload
::EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiSubscriptionResponseType
EiManagedSegmentReferenceDataPayload
::EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL



Deleted:

Figure 13-7 UML Class Diagram of Payloads for Segment Reference Data Subscriptions

The subscription payloads for delivery of Session Reference Data is a single Segment Reference Data Type object (Figure 13-6 and Table 13-5 Segment Reference Data).

13.4 Trading Session Data

The Market Structure Report tells the Party how to trade. Following the classification used by FIX, Market Structure Reference Data is just one part of Pre-Trade Information.

Segment Reference Data includes information on Opening and Closing Auctions, as well as information for specific Instruments. It also includes detailed information to guide trading, negotiation, execution, settlement (not in scope for CTS) and audit (supported by CTS, wherein the difference between market Position and measured Delivery is automatically executed).

Deleted: Structure

Deleted: ,

Deleted: Crossing

Deleted: and

Deleted: settled

13.4.1 Messages for Trading Session Data

Table 13-6 Messages for Trading Session Data

Facet	Request Payload	Response Payload	Notes
Reference Data	EiManage Session Data Payload	EiManaged Session Data Payload	Messages are subclasses of the Subscription Management Messages

13.4.2 Interaction Pattern for Trading Session Data

Trading Session Data is very dynamic, and includes (e.g.) information on planned and unplanned closures, auctions, and more. It follows the Subscription pattern—see Section 10 *"Subscription Facet."*

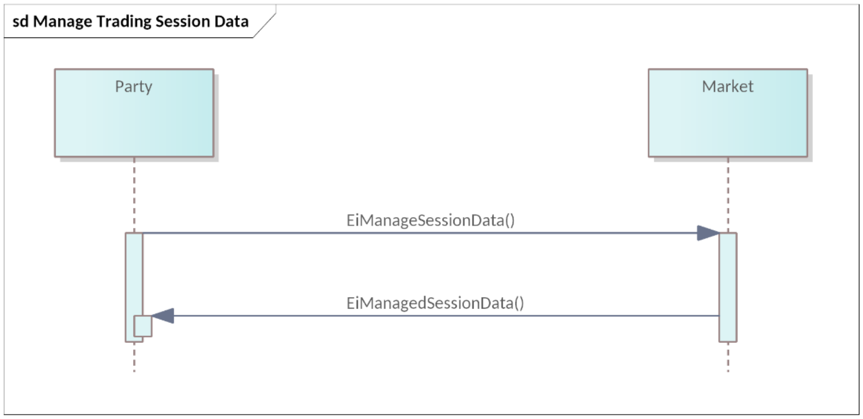
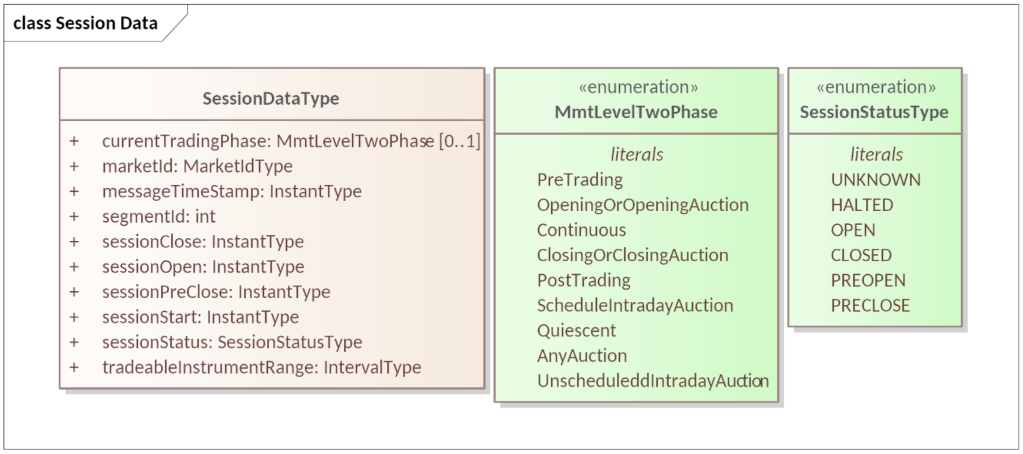


Figure 13-8: UML Sequence Diagram for Manage Trading Session Data

A Party may watch changes to a single Session by naming that Session in the subscription request. This will return only that Session and updates to that Session.

13.4.3 Information Model for Trading Session Data

Figure 13-9 shows the UML Class Diagram for Session Data and the enumerations used.



Deleted:

Figure 13-9 UML Class Diagram for [Trading] Session Data

The attributes for Session Data are in Table 13-7.

Table 13-7 Session Data

Attribute	Attribute Type	FIX Field	Meaning	Comments
Current Trading Phase	MMT Level Two Phase Enumeration	TradingSessionSubID (625)	Active trading phase for the session. The Enumeration is based on the description of FIX Codes 1 through 9 and describes the same trading phases.	The values used in CTS are • PreTrading • Opening Or Opening Auction • Continuous [trading] • Closing Or Closing Auction • PostTrading • Scheduled IntraDay Auction • Quiescent • Any Auction • Unscheduled Intraday Auction
Market ID	Market ID Type	MarketID (1301)	Identifies the containing market	
Message Time Stamp	Instant Type	TransactTime(60)	The timestamp for when the Session Data was produced.	
Segment ID	Integer	MarketSegmentID (1300)	Identifies the containing Segment	This is unique when paired with the Market ID
Session Close	Instant Type	TradSesCloseTime (344)	Closing time of the trading session	Session times may vary for different Market Mechanisms (Section 0)
Session Open	Instant Type	TradSesOpenTime (342)	Time of the opening of the trading session	Follow FIX semantics.
Session PreClose	Instant Type	TradSesPreCloseTime (343)	Time of the pre-close of the trading session	Follow FIX semantics.
Session Start	Instant Type	TradSesStartTime (341)	Starting time of the trading session	Follow FIX semantics.
Session Status	Session Status Type Enumeration	TradSesStatus (340)	The status of this session	Values are • Unknown • Halted • Open • Closed • PreOpen • PreClose

Deleted: Not in FIX

Deleted: (from FIX code set)

Attribute	Attribute Type	FIX Field	Meaning	Comments
Tradeable Instrument Range	Interval Type	Not in FIX	Instruments whose Interval is contained in the Tradeable Instrument Range may be traded.	See Section 3.3 “The Bounding Interval Pattern in CTS” Segments have their own Tradeable Instrument Range which may be less dynamic.

13.4.4 Payloads for Trading Session Data

The payloads for Session Data are those of the Subscription Management Messages. See Figure 13-10.

The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.

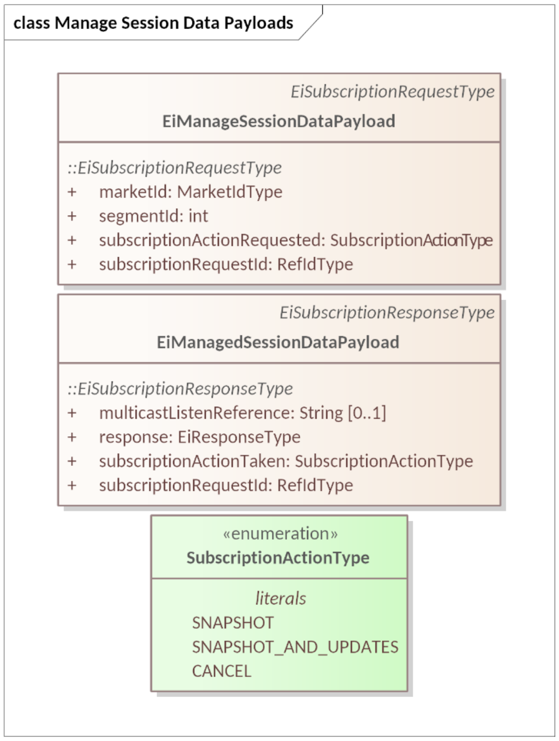

class Manage Session Data Payloads
EiSubscriptionRequestType
EiManageSessionDataPayload
::EiSubscriptionRequestType
+ marketId: MarketIdType
+ segmentId: int
+ subscriptionActionRequested: SubscriptionActionType
+ subscriptionRequestId: RefIdType
EiSubscriptionResponseType
EiManagedSessionDataPayload
::EiSubscriptionResponseType
+ multicastListenReference: String [0..1]
+ response: EiResponseType
+ subscriptionActionTaken: SubscriptionActionType
+ subscriptionRequestId: RefIdType
«enumeration»
SubscriptionActionType
literals
SNAPSHOT
SNAPSHOT_AND_UPDATES
CANCEL

Figure 13-10 UML Class Diagram of the Payloads for Manage Session Data

Deleted:

14 Conformance

14.1 Introduction to Conformance

By design, CTS is a simplified and restricted subset profile of TeMIX with extension for financial market semantics. See Appendix D.

Deleted: .

Portions of CTS conform to, and use updated and simplified versions of the specifications consumed by EI, specifically:

- OASIS WS-Calendar **[WS-Calendar]**
- A definition of Streams contained in **[EI]**

We normatively reference and apply the evolution of these specifications, in particular:

- OASIS WS-Calendar Schedule Streams and signals **[Streams]**, simplified as CTS Streams.
- The WS-Calendar **[CAL-MIN]** Interval is used directly (as IntervalType).

This specification simplifies WS-Calendar Schedule Streams and Signals [Streams] as CTS Streams (see Section 0) and refactors the TEMIX profile of **[EI].**

14.2 Claiming Conformance to Common Transactive Services

Deleted: Conformance of the CTS evolved specification can be shown with the techniques of **[IEC62746-10-3]** is described in informative .¶

Implementations claim conformance to Common Transactive Services 1.0 by asserting conformance statements on the numbered items below.

For an implementation to conform

1) The conformance statement MUST list all Facets which it supports in full or and in part. The list MUST include the Tender and Transaction Facets and also Sections 2.4 and 2.5.
2) Stream Tenders and Quotes need not be supported, but whether the implementation supports them MUST be described along with any limitations on their use.
3) The conformance statement MUST describe all extensions to payloads described in this specification.
4) The conformance statement MUST describe the serialization Binding(s) which the conforming implementation uses along with bindings for any extensions to CTS. Note that at present there are no standardized bindings in this specification.[29]
5) The conformance statement MUST describe how each payload definition conforms to the UML and/or profiled definitions for each payload unless it uses only standard Bindings planned for a future Technical Committee document.
6) The conformance statement MUST indicate cardinality for message payload attributes where there is flexibility in this specification.
7) The conformance statement MUST describe any facets it defines to extend this specification.
8) The conformance statement MUST describe how and to what extent the implementation supports the Subscription Facet (Section 10) and MUST include a list of supported Tickers (Section 11).
9) The conformance statement MUST describe how and to what extent Session, Instrument, Market, Segment, and Instrument Data and subscriptions are produced and delivered (Sections 12 and 13).
10) The conformance statements MUST describe what Market Mechanisms are presented to users of the implementation from the list in Section 13.1 Market Mechanisms and Appendix B.

Deleted: it supports

Deleted: . If the implementation does not use a standard binding as defined in Section 13, the conformance statement MUST define the binding used, at a similar level

Deleted: detail to Section 13.

Deleted: in Section 13

Deleted: Annex: Conformance

Deleted: Spec not yet incorporated into

14.3 FIX Conformance

Wherever possible this specification uses concepts and terminology defined by the FIX Trading Community (https://www.fixtrading.org/) as expressed in their online standards support tool **[FIXIMATE]**.

Deleted: section

[29] Formal bindings are planned by the Technical Committee in a forthcoming Technical Report *Bindings for Common Transactive Services (CTS) Version 1.0.*

All words and terms in the tables that reference a FIX Field are as defined by FIX Trading Community standards and referenced through FIXimate. All have the meaning and effect in interactions as specified by the FIX Trading Community.

CTS and the broad range of FIX specifications have different goals—CTS is a simplified standard means of interacting with markets, and typically does not express the full richness of complex markets—so CTS has things that are not in FIX and *vice versa*.

There are some differences in naming for operations; FIX naming convention for messages and operations uses the term “Request” in the message name to invoke an operation. A field inside the request message contains the desired operation, e.g. “Cancel”.CTS instead uses siimple strongly typed messages that are less complex (and less expressive of market complexities).

Deleted: <#>Conforming with Use of Warrants in Tenders¶

14.4 Warrants in Tenders

Warrants increase the specificity of Product (and Instrument). A Buyer who does not specify a Warrant will be satisfied by Delivery of a Product whether or not it has a Warrant. A Buyer who requests Product with a Warrant will only be satisfied by Delivery of a Product that has that Warrant.

Consider a buyer who wishes to buy a package of coffee beans and a buyer who wishes to buy a package of organic coffee beans. The word “Organic” on the label serves as a Warrant. The first buyer will buy solely on price, and is indifferent to seeing the word “Organic” on the label. The second buyer will choose only from among those packages with the warrant “Organic” on the label.

When a Tender on the Buy side specifies a Warrant, it must be rejected by any Market Segment that does not include that Warrant. A Tender on the Sell side that specifies a Warrant may be accepted by any Segment where the same Resource and Duration are traded. Conversely, a Tender on the Sell side without a Warrant must be rejected by any Segment that specifies a Warrant.

Note that warrants are identified by WarrantID (see Sectioni 2.5). Comparison methods MUST be descriibed ini the conformance statement. It suffices to compare the WarrantIDs, which are subclassed UIDs.

Appendix A. References

This appendix contains the normative and informative references that are used in this document. Normative references are specific (identified by date of publication and/or edition number or Version number) and Informative references may be either specific or non-specific.

While any hyperlinks included in this appendix were valid at the time of publication, OASIS cannot guarantee their long-term validity.

A.1 Normative References

The following documents are referenced in such a way that some or all of their content constitutes requirements of this document.

[CAL-MIN]
WS-Calendar Minimal PIM-Conformant Schema Version 1.0. Edited by William Cox and Toby Considine. 26 August 2016. OASIS Committee Specification. http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html

[CAL-PIM]
OASIS WS-Calendar Platform-Independent Model version 1.0, Committee Specification 02 Edited by William T. Cox and Toby Considine, 21 August 2015. http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/cs02/ws-calendar-pim-v1.0-cs02.html Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html

[EI]
Energy Interoperation Version 1.0. Edited by Toby Considine, 11 June 2014. OASIS Standard. http://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html Latest version: http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html. and its TeMIX Profile

[EMIX]
OASIS Energy Market Information Exchange (EMIX) Version 1.0 Committee Specification 02 Edited by Toby Considine, 11 January 2012. http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html Latest version: http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html

[JSON]

JavaScript Object Notation and JSON Schema. https://cswr.github.io/JsonSchema/

[**MMT]**
FIX Trading Community Market Model Typology v4.2, retrieved July 2, 2024, https://www.fixtrading.org/mmt/

[RFC8174]
Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <http://www.rfc-editor.org/info/rfc8174>.

[RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2246]
T. Dierks, C. Allen *Transport Layer Security (TLS) Protocol Version 1.0*, http://www.ietf.org/rfc/rfc2246.txt, IETF RFC 2246, January 1999.

[SBE]
Simple Binary Encoding Technical Specification 1.0. FIX Trading Community, June 16, 2016. https://www.fixtrading.org/standards/sbe/

[Streams]
Schedule Signals and Streams Version 1.0. Edited by Toby Considine and William T. Cox. 18 September

2016. OASIS Committee Specification. http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html.

A.2 Informative References

The following referenced documents are not required for the application of this document but may assist the reader with regard to a particular subject area.

[Actor Model]
C. Hewitt, "Actor Model of Computation: Scalable Robust Information Systems," arxiv.org, 2010.

[Fractal Microgrids]
Art Villanueva et al, *Camp Pendleton Fractal Microgrid Demonstration,* California Energy Commission Report CEC-500-2016-013,j available at http://400.sydneyplus.com/CaliforniaEnergy_SydneyEnterprise/Download.aspx?template=Books&field=PublicURL&record=57483797-a40e-49e7-b675-2858a3ad0d91&showSave=False&repeat=d4e63b56-27d1-4476-9300-7ede86a533ca

[Framework]
National Institute of Standards and Technology, *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0*, January 2010, http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf

[CTS2016]
W.T. Cox, E. Cazalet, E., A Krstulovic, W Miller, & W.Wijbrandi *Common Transactive Services*. TESC 2016. Available at http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf

[EML-CTS]
Energy Mashup Lab Common Transactive Services (open-source software) https://github.com/EnergyMashupLab/eml-cts)

[FIXIMATE]
FIXimate^{sm} FIX Interactive Message And Tag Explorer
https://fiximate.fixtrading.org/

Deleted: FIXimate

[FSGIM]
Facility smart grid information model. ISO 17800. https://www.iso.org/standard/71547.html 2017

[iCalendar]
B. Desruisseaux*, Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, https://tools.ietf.org/html/rfc5545. 2009,
See also
C. Daboo & M. Douglas. *Calendar Availability*, https://tools.ietf.org/html/rfc7953, 2016

[GridFaultResilience]
W.T. Cox & T. Considine. *Grid Fault Recovery and Resilience: Applying Structured Energy and Microgrids*. IEEE Innovative Smart Grid Technologies 2014. Available at http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf

[**IEC62746-10-3]** International Standard.
Systems interface between customer energy management system and the power management system - Part 10-3: Open automated demand response - Adapting smart grid user interfaces to the IEC common information model, https://webstore.iec.ch/publication/59771 2018.

[Micromarkets]
W.T. Cox & T. Considine, *Energy, Micromarkets, and Microgrids.*
GridInterop 2011, https://www.gridwiseac.org/pdfs/forum_papers11/cox_considine_paper_gi11.pdf

[RFC3552]
E Rescorla & B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[SmartGridBusiness]
T. Considine & W.T. Cox, *Smart Loads and Smart Grids—Creating the Smart Grid Business Case*. Grid-Interop 2009. Available at http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf

[StructuredEnergy]
Structured Energy: Microgrids and Autonomous Transactive Operation, http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-Cox_StructuredEnergyPaper518.pdf. Innovative Smart Grid Technologies 2013 (IEEE).

[TPB-EI]
Transport Protocol Bindings for OASIS Energy Interoperation 1.0 Version 1.0. 01 October 2012. OASIS Committee Specification 01. https://docs.oasis-open.org/energyinterop/tpb-ei/v1.0/tpb-ei-v1.0.pdf .

[TeMIX]
TeMIX Transactive Energy Market Information Exchange [TeMIX] an approved Note of the EMIX TC. Ed Cazalet et al., 23 May 2010. http://www.oasis-open.org/committees/download.php/37954/TeMIX-20100523.pdf

[TransactiveMicrogrids]
Jennifer M. Worrall, Edward G. Cazalet, PhD, William T. Cox, PhD, Narayanan Rajagopal, Thomas R. Nudell, PhD, and Paul D. Heitman, *Energy Management in Microgrid Systems,*TESC 2016. Available at http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/tes2016_microgrids_paper_Final.pdf

[TRM] (Transactive Resource Management)
B. Huberman and S. H. Clearwater, *Thermal markets for controlling building environments*, Energy Engineering*,* vol. 91, no. 3, pp. 26- 56, January 1994.

[UML]
Object Management Group, *Unified Modeling Language (UML), V2.5.1*, December 2017. https://www.omg.org/spec/UML/2.5.1/

Deleted: *4.1*, August 2011. http://www.omg.org/spec/UML/2.4.1/

[XSD]
W3C XML Schema Definition Language (XSD) 1.1. Part 1: Structures, S Gao, C. M. Sperberg-McQueen, H Thompson, N Mendelsohn, D Beech, M Maloney http://www.w3.org/TR/xmlschema11-1/, April 2012, Part 2: Datatypes, D Peterson, S Gao, A Malhotra, C. M. Sperberg-McQueen, H Thompson, P Biron, http://www.w3.org/TR/xmlschema11-2/ April 2012

[ZeroTrust]
Zero Trust Architecture, S Rose, O Borcher, S Mitchell, S Connelly. NIST Special Publication 800-207, https://doi.org/10.6028/NIST.SP.800-20 August 2020

Appendix B. Choosing a Market Mechanism

A Market may consist of several segments. Segments differ chiefly in the Products traded and in the Mechanism they use for trading. Market Participants will select different Segments based in part on how well the Market Mechanism in that Segment supports its goals.

The authors of CTS cannot specify which market mechanism to use in every situation. This non-normative section discusses each named Market Mechanism and describes some scenarios in which it might be used.

See Section 13.1 **“**Market Mechanisms**”** for the normative description of market mechanisms.

B.1 Central Limit Order Book (LB): Simple Bids & Offers

The central limit order book, also known colloquially simply as an order book, is the simplest market for many to understand. Would-be buyers submit bids which the market immediately executes or writes in the book. Would-be sellers submit offers which the market writes in the book. The order book represents the collective actions of buyers and sellers who place orders to buy or sell an asset at a specific price.

The order book continuously updates as new orders are added or existing ones are matched or canceled. Transactions are created when the Segment matches compatible buy orders and sell orders. Matches are made based on price-time priority, where orders are filled based on the highest bid and the lowest ask, and when these prices overlap, the order that was placed first gets priority.

When a submitted order is matched against one or more orders on the book, trades are executed, and the order book is updated to reflect the new supply and demand levels.

In summary, the order book market matching process is a continuous cycle of order formation, order matching, and price discovery, driven by the interactions of buyers and sellers in the marketplace.

B.2 Periodic Auctions (PA)

Participants in periodic auctions submit bids and offers up until a published deadline. After the deadline, all tenders are evaluated and a common price is determined, e.g. at which the greatest volume can be executed. All bids (tenders to buy) above that common price are accepted, and all offers (tenders to sell) below that price are accepted. All transactions clear at the common price. Any remaining Tenders are referred to as the Residual.

The North American bulk power markets are run largely through periodic auctions (e.g., with transactions in day-ahead markets announced the day before) to enable large generators to schedule their operations.

Periodic auctions are also referred to “Double Auctions”, that is, auctions in which both sellers and buyers submit bids.

B.3 Quote-Driven Markets (QB)

Quote-driven markets typify markets with dominant suppliers or market makers providing additional liquidity by offering to buy and/or sell at any time. The price of the resource is determined and announced by the dominant suppliers. The dominant suppliers MAY represent many third parties, i.e., a distribution system operator (DSO) acting as an intermediary to a bulk power market.

Quote-Driven Markets permit partial lifting, that is a Party may issue a Tender for 7kWh in response to a Quote for 150 kWh. Quote Driven Markets do not normally process Rejections or accept counter-Quotes as a Quote Response; Quotes are issued on a take-it-or-leave-it basis.

In typical FIX Protocol-based markets, quotes are non-negotiable and are valid for a short time, perhaps three seconds. In existing power markets using quotes for day-ahead markets, they may be available for hours. In either case, if buyers take all of the quantity in a quote, a dominant supplier has the option of submitting a new quote, potentially at a different price.

A common example of a quote-driven market is one in which an electric utility announces 24 hourly prices for the next day. These are good until a certain time and indicate the maximum quantity that the issuer is willing to sell at that price. In CTS, these will be tradeable quotes. A buyer or seller submits a Quote Response containing a Tender which can automatically match against a tradeable quote.

This pattern is useful because at the limit, all resource markets are limited by the maximum potential rate of resource delivery. Once that limit is reached, the market maker may avoid all further transactions by not entering additional quotes.

In some regulated electricity markets, the price received for selling power is less than the price paid for buying power, establishing a spread. A market maker may opt to publish a range of quotes to buy as well and benefit from the spread by buying and selling a resource at different times as the prices fluctuate.

Quote driven CTS markets are typically highly regulated markets.

B.4 Request for Quote Markets (RQ)—Negotiating

Request for quote markets support bilateral negotiations around price and quantity. Interactions begin with a request for quote (RFQ), which may be vague as to prices, quantities, or even the price schedule of the Instruments. The recipient of an RFQ may reply with one or more indicative or tradeable quotes, perhaps for different delivery times or quantities. When an acceptable tradeable Quote is received, the party wishing to lift that Quote responds with a Quote Response that notifies the Market of the acceptance. The Market then generates a Transaction.

The Negotiation process is inherently flexible. A Transaction may come after many rounds of negotiation, or directly from a response to the first tradeable quote. This section describes some potential interactions to clarify the concepts.

An RFQ Market can permit large buyers to plan significant resource use over time, for example, scheduling a long running industrial process which also requires labor planning. Such a buyer could submit multiple Requests for Quotes with different schedules, and then select from among the Quotes received in response.

An RFQ uses a Bounded Interval to indicate what an acceptable Quote would be.

- Consider Party A that wishes to buy 15 kW of power over a two-hour period, sometime within an 8-hour window. This would take the form of submitting an RFQ to Party B with an eight-hour Bounding Interval with a specific start time, but with a Stream of two Intervals with a Duration of 1 hour but with no starting time specified.
- Consider instead that Party A further wishes to buy 10 kW of energy over an hour at $0.05/kWh sometime during the work day. Party A can issue an RFQ, with a bounding duration of the workday, containing a single unscheduled Interval of one hour containing the Price and the Quantity.

Party A and B can send these RFQs directly to one or more potential counterparties or published to the entire market. Because it is not tradeable, a RFQ does not need to be submitted to the Market and the Segment does not need to register the RFQ. The response is a quote, either indicative or tradeable.

Party A may receive one or more offers in response. These become more specific, perhaps two Quotes issued by the same counterparty with different prices at different times. A quote issuer may make a counteroffer by sending a quote proposing different quantities and/or prices. Perhaps the responding Party considers that it will turn on a generator, but only if it can operate the generator at an economic rate for an economic duration. A quote MAY be for only one of the two hours indicated in the original RFQ, leaving the requesting Party to find an acceptable match from among all the offers (quotes) it receives. The prices may be higher or lower than requested in the original RFQ. Until one party issues a tradeable Quote, all responses are RFQs or indicative Quotes, issued to continue the negotiation.

When either Party thinks that there is an essential meeting of requirements, that Party submits a tradeable Quote, that is a quote that a matching Tender will turn into a Transaction. For a CTS quote to be tradeable, the Party informs the Segment, even if it is a private quote and not published.

To accept a tradeable Quote, a Party submits a Tender to the Segment, referencing the Quote ID, and matching the details of the quote. The market mechanism compares the Tender to the quote, and, if they match, it executes the Transaction. All tradeable quotes are treated as if they are marked All-or-None (AON).

The issuer normally accepts a Tender received in response to a tradeable Quote, with exceptions for expiration or for another Tender having gotten to the Segment first.

The issuer MUST accept a Tender received in response to a tradeable Quote.

Negotiations may include Interval Quotes or Stream Quotes, a pattern that matches that of Tenders (See Section 5.3.1, *"Interval Tenders and Stream Tenders."*) A Stream Quote must be matched to a Stream Tender in the Quote Response to create a Transaction. A requester that wishes to acquire a power curve indicates this with a Stream Quote back indicates it in the RFQ or indicative Quote. Note: the response does not need to match the request; the Indicative or Tradeable Quote received in response may propose a different Stream.

Below are three non-normative scenarios for negotiation to illustrate the flexibility of Negotiation: (1) Single-provider, (2) over-the-counter, and (3) system recovery.

1) Parties may choose to use an negotiated trade because they wish to bypass certain market restrictions. For example, consider a Party wishing to buy 87 kWh of Power for a period over 1 hour and 5 minutes beginning 15 minutes after the hour. Parties negotiate as above, come to terms, as above, and the Segment records the Transaction.
   Order Book Segments impose restrictions on Round Lots and Intervals to improve Market Liquidity, that is, the likelihood of a match between a Tender to Buy and a Tender to Sell. If Parties already have made an agreement, then there is no need to improve liquidity. This makes the Durations and Round-Lots in negotiated markets indicative rather than prescriptive. 87 kWh is a rough match for Off-Market Segment with a 20 kWh round lot—a gWh is not. In a comparable way, the quote's Duration is a rough match for Segment with an Hour Duration while 3 minutes is not.
2) Markets commonly project opening prices for Instruments before they open. If a system recovery requires a market re-start, there may be no good information to set opening prices. Prior to a re-start, a Segment may publish RFQs to buy and to sell. The Operator may use a Segment to probe the potential market in this way several times, perhaps with different prices and quantities, to discover an indicative opening price at which the Resource will be in rough balance. (This rough balancing MAY be by an implied auction.) When the market has enough information, then the market opens a Segment for trading, announcing the indicative opening prices for each Instrument.

This specification does not require that a Market segment include any of the scenarios described above. We include them to illustrate how the essential components of Negotiation might fit together in a specific system.

B.5 Market mechanisms not defined in FIX MMT

As traditional regulated and centrally-managed markets migrate to TE, CTS supports some mechanisms not defined in the FIX Market Model Typology (MMT).

B.5.1 Off-Book segment (OB)

Off Book mechanisms are reserved in CTS for direct allocation of Resource from one Party to another by a process external to the Market. Parties are notified through the receipt of trade notices (Transactions).

A transactive resource market may be used to balance resource flows within a microgrid or other local distribution system. Markets solve the knowledge problem of balancing supply and demand over time even when all parties or systems have a common owner.

A common scenario, say on a campus or base, is to handle scarce resources through direct assignment to one of the parties. Consider a campus of 10 buildings and a hospital. The owner may wish to create a Transaction in which each of the buildings transfers 10 kWh to the hospital which receives 100 kWh into its position. The donor buildings must then trade within their own accounts to rebalance supply and demand, or re-balance operations to stay within their new position.

On military bases, this can be referred to as power following command intent.

B.5.2 Real Time Pricing (RT)

Price quotes are broadcast for each Interval, but the Segment has no mechanism for negotiations or Tenders. Transactions are generated later by reading Delivery and generating transactions in a self-executing Segments.

B.5.3 Spot Market (SP)

A Ticker in a spot market indicates the “instant” price in the Segment indicating the Price for purchases or sales. A spot market Segment may limit active trading to a small window of time.

A spot market segment MAY accept Tenders to sell as the market maker tries to pull a resource back into the market to address a looming shortfall.

A spot market may support an asymmetry of self-execution, perhaps creating transactions for un-planned consumption but not for un-planned sales.

B.5.4 Self Executing (SX)

A self-executing Segment creates Trades to align with consumption as measured at the meter and reported by the Delivery facet. Self-execution generates Transactions for Resource consumed without previously being bought. Self Execution aligns the Position known to the Market with the amount consumed as indicated by Delivery.

A self-executing Segment is needed to augment any other market mechanism so long as the Resource delivered to a customer is not limited to the amounts transacted in advance; in today’s power markets, the power delivered will not be limited to the customer’s market position.

Appendix C. Security and Privacy Considerations

This specification defines message payloads only. Security must be composed in. Privacy considerations must be decided when implementing specific systems for specific purposes.

C.1 CTS and Security Considerations

Procuring energy for local use and selling energy for remote use are each at the cusp of finance and operations.

- A price that is falsely low may cause the buyer to operate a system when there is inadequate power, potentially harming systems within a facility, or harming other facilities on the same circuit.
- A price that is falsely low may cause the seller to leave the market.
- A price that is falsely high may cause the buyer to shut down operation of systems or equipment.
- A price that is falsely high may cause the seller to increase operations when there is neither a ready consumer nor perhaps even grid capacity to take delivery.

For these reasons, it is important that each system guard the integrity of each message, assure the identities of the sender and of the receiver, and prove whether a message was received or not.

Messages should be encrypted to prevent eavesdropping. Any node should be able to detect replay, message insertion, deletion, and modification. A system must guard against and detect man-in-the-middle" attacks wherein an intermediary node passes of messages as originating from a known and trusted source.

The Technical Committee generally recommends that production implementations use Zero-Trust security **[ZeroTrust]**, especially because of the wide distribution and potentially diverse ownership of TRM Actors. Zero Trust security requires authentication and authorization of every device, person, and application. The best practice is to encrypt all messages, even those between the separate components of an application within the cloud.

This specification makes no attempt to describe methods or technologies to enable Zero Trust interactions between Actors.

C.2 CTS and Privacy Considerations

The United Nations has defined privacy as “the presumption that individuals should have an area of autonomous development, interaction and liberty, a ‘private sphere’ with or without interaction with others, free from state intervention and excessive unsolicited intervention by other uninvited individuals. The right to privacy is also the ability of individuals to determine who holds information about them and how that information is used” (UN General Assembly 2013:15).

Electrical usage data inherently creates a privacy risk. Published work has demonstrated that simple usage data can be used to reveal the inner operations and decisions in a home. Other research has demonstrated that anonymous electrical usage data can be “de-anonymized” to identify an individual electricity user. The more fine-grained the data, the more intimate the details that can be garnered from meter telemetry.

In an amicus brief in a case on smart metering, the Electronic Freedom Foundation testified that that aggregate smart meter data collected from someone’s home in 15-minute intervals could be used to infer, for example, whether they tend to cook meals in the microwave or on the stove; whether they make breakfast; whether and how often they use exercise equipment, such as a treadmill; whether they have an in-home alarm system; when they typically take a shower; if they have a washer and dryer, and how often they use them; and whether they switch on the lights at odd hours, such as in the middle of the night. And these inferences, in turn, can permit intimate deductions about a person's lifestyle, including their occupation, health, religion, sexuality, and financial circumstances. These privacy concerns are linked to increased security risks criminals may be able to access the data and use the information to enable inferences about what people are doing in their home or if they are away from home.

This specification describes how to share communications beyond mere electrical usage telemetry. Communications reveal what the user would like to buy, how much they would be willing to spend, and future intents and plans.

System developers using this specification should consider legal requirements under the Fair Practice Principles and the European Union's General Data Protection Regulation. These include:

1) The Collection Limitation Principle. There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject.
2) The Data Quality Principle. Personal data should be relevant to the purposes for which they are to be used and, to the extent necessary for those purposes, should be accurate, complete and kept up to date.
3) The Purpose Specification Principle. The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.
4) The Use Limitation Principle. Personal data should not be disclosed, made available or otherwise used for purposes other than those specified, except a) with the consent of the data subject, or b) by the authority of law.
5) The Security Safeguards Principle. Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorized access, destruction, use, modification or disclosure of data.
6) The Openness Principle. There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data and the main purposes of their use, as well as the identity and usual residence of the data controller.
7) The Individual Participation Principle. An individual should have the right:

to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him.

to have data relating to him communicated to him, within a reasonable time, at a charge, if any, that is not excessive; in a reasonable manner, and in a form that is readily intelligible to him.

to be given reasons if a request made under subparagraphs (a) and (b) is denied and to be able to challenge such denial; and

to challenge data relating to him and, if the challenge is successful, to have the data erased, rectified, completed or amended.

8) The Accountability Principle. A data controller should be accountable for complying with measures which give effect to the principles stated above.

In developing this specification, the Technical Committee has kept in mind the need to support a developer wishing to support privacy. Actors representing an up-stream electrical serving entity, say a distribution system operator or traditional utility, use the tame messages as anyone else—no actor is inherently privileged. Messages to provide market information or “tickertape” functions do not include Party IDs. General advertising of Tenders, while necessary to draw matching Tenders quickly to market, may be anonymous.

In some messages and some markets, it is necessary to use a proxy ID to protect privacy or to simply conveyance of a transaction from a complex matching mechanism. To protect privacy, a market may transmit such a proxy ID in place of a Party Id in Quotes, Tenders, Transactions, and Tickers. Markets that use cumulative matching algorithms such as double auction cannot identify a specific Counter Party to a transaction.

The system developer should keep the privacy principals in mind when making specific technology choices. For example, messages between an actor and the market MAY be encrypted to protect the privacy of people represented by individual actors. While the transactive energy market must know both buyers and sellers to support transactions and settlements, the developer should take steps to guard that information. A developer may opt that each notice of contract sent to an actor always has a counterparty of the market, so as to protect the sources and uses of electricity.

It is beyond the scope of this specification to specify security practices and privacy design for markets built using this specification.

Appendix D. Semantic Composition from and relationship to Energy Interoperation, EMIX, and WS-Calendar

This Common Transactve Services specification draws on semantics previously defined by this and related Technical Committees. The history presented in this informative Appendix may be of interest to readers.

The predecessor standard, Energy Interoperation [EI] of which the TEMIX profile is part, applies other standards, including [EMIX] and [WS-Calendar], and uses an earlier Streams definition. We have adapted, updated, and simplified the use of the referenced standards, while maintaining semantic conformance in a broad sense.[30]

Deleted: The semantics and interactions of CTS are selected from and derived from [EI]. ¶
EI references two

Deleted: ,

Deleted: adapt, update

Deleted: simplify

Deleted: .

We note that

- EMIX described price and product for electricity markets.
- WS-Calendar communicates schedules and sequences of operations. CTS uses the [Streams] optimization, which is a standalone specification, rather than part of EI 1.0.
- EI uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each EI service references a product defined using **[EMIX]**. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EI] is expressed using [WS-Calendar].
- Since [EI] was published, a semantically equivalent but simpler [Streams] specification was developed in the OASIS WS-Calendar Technical Committee. CTS uses that simpler [Streams] specification.

Deleted: describes

All terms used in this specification are as defined in their respective specifications.

In [EI], the fundamental resource definition was the [EMIX] Item, composed of: a resource name, a unit of measure, a scale factor, and a quantity. For example, a specific EMIX Item may define a Market denominated in 25 MWh bids. [EI] defined how to buy and sell items during specific intervals defined by a duration and a start time. The Quotes, Tenders, and Transactions that are the subject of [EI] added specific prices and quantities to the item and interval. EMIX optionally included a location, i.e., a point of delivery for each [EI] service.

In CTS, we group and name these elements as a Resource, Product, and Instrument. These terms are defined in Section 2.2.4, “Markets and Market Segments”

Note that the informational elements in a fully defined tender or transaction are identical to those described in EMIX. The conceptual regrouping enables common behaviors including Market discovery and interoperation between Actors built on different code bases.

D.1 Conformance with Energy Interoperation

EI defines an end-to-end interaction model for transactive services and for demand response. CTS uses the EI transactive services and draws definitions of parties and transactive interactions primarily from the [EI] TEMIX profile.

This specification can be viewed as a minimal transactive profile of [EI].

[30] Conformance of the CTS evolved specification to these previous specifications can be shown with the techniques of informative **[IEC62746-10-3]** .

D.2 Conformance with EMIX

This specification uses a simplified profile of the models and artifacts defined in OASIS Energy Market Information Exchange **[EMIX]** to communicate Product definitions, quantities, and prices. EMIX provides a succinct way to indicate how prices, quantities, or both vary over time.

The EMIX Product definition is the Transactive Resource in CTS 1.0.

EMIX defines Market Context, a URI used as the identifier of the Market. EMIX further defines Standard Terms as retrievable information about the Market that an actor can use to configure itself for interoperation with a given Market. We extend and clarify those terms, provide an extension mechanism, and discuss the relationship of markets, market segments, and products.

D.3 Conformance with WS-Calendar Streams

WS-Calendar expresses events and sequences to support machine-to-machine (M2M) negotiation of schedules while being semantically compatible with human schedules as standardized in [**iCalendar**]. Schemas in **[WS-Calendar]** support messages that are nearly identical to those used in human schedules. We use a conformant but simpler and more abstract Platform Independent Model **[CAL-PIM]** and the [**Streams**] compact expression[31], to support telemetry (Delivery Facet) and series of Tenders while not extending the semantics of [**Streams**].[32]

WS-Calendar conveys domain specific information in a per-event payload within a schedule-centric message; in CTS, the domain is the price, product, and quantity. An essential concept of WS-Calendar is inheritance, by which a starting time can be applied to an existing message, or by which all events in a sequence share common information such as duration. Inheritance is used to “complete” a partial message during negotiation. CTS makes use of this to apply a common product across a sequence, or to convey a specific starting time to a market product.

D.4 CTS and WS-Calendar Streams

The **[Streams]** specification describes how to handle repeating time series of similar data, applying repeating information to a series of schedulable intervals, expressing common information once for the series, overriding the common information only if needed within a specific interval, and potentially scheduling (“binding”) the entire series by adding a starting date and time to one of the Intervals.

For CTS, this means that a Product is fully described in the header, and only the elements that vary, such as the Price or the Quantity, are expressed in the intervals.

CTS Streams use this same format even when the Intervals contain only a single Interval.

In addition, CTS Streams include energy-market elements that are outside the Streams standard but follow the pattern of referrals as defined in [Steams] conformance.

CTS Streams have neither interaction patterns nor payloads, as they are a common abstract information model used to define the messages used in Facet messages.

The CtsStream follows this pattern. The elements from **[Streams]** have been flattened into the CTS Stream, and the Stream Interval and payload flattened into a streamPayloadValue and the internal local UID for the stream element.

[31] Simplified as CTS Streams in this specification.

[32] Some specifications (e.g. [FSGIM]) have extended the basic [Streams] capabilities, but this brings additional complexity which does not benefit our use cases.

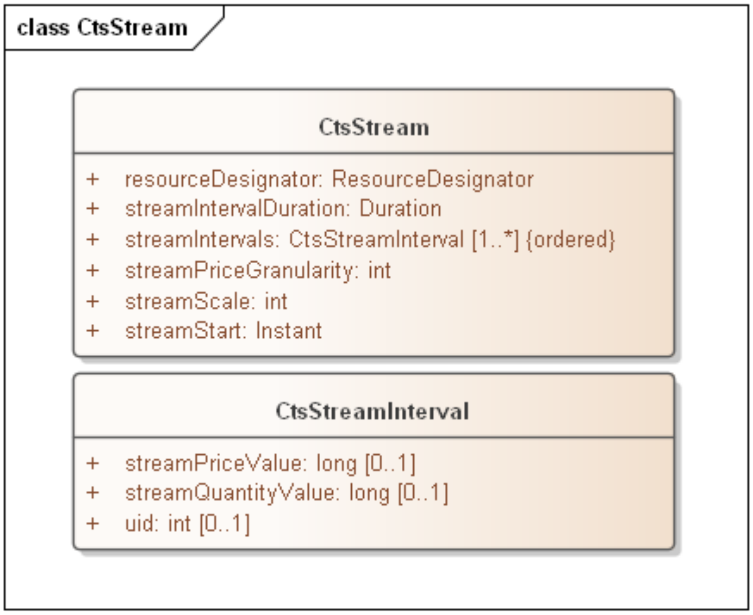

class CtsStream
CtsStream
+ resourceDesignator: ResourceDesignator
+ streamIntervalDuration: Duration
+ streamIntervals: CtsStreamInterval [1..*] {ordered}
+ streamPriceGranularity: int
+ streamScale: int
+ streamStart: Instant
CtsStreamInterval
+ streamPriceValue: long [0..1]
+ streamQuantityValue: long [0..1]
+ uid: int [0..1]

Figure D-14-1: CtsStreamDefinition

Deleted: *C*

As with **[Streams]**, CtsStreamIntervals are ordered, that is the sequence of intervals is essential. Some serialization specifications, notably XML, do not require that order be preserved when deserializing a list. The UID enables proper ordering of the Stream Intervals if order is not preserved. Since conformant CTS implementations need not be owned by the same implementer, and may pass through multiple translations, the UID property is required.

Appendix E. Glossary of Terms and Abbreviations Used in this document

Throughout this document, abbreviations are used to improve clarity and brevity, especially to reference specifications with long titles.

Table C--14-1 Abbreviations and Terms used throughout this document

Attribute	Meaning
CTS	Common Transactive Services
EI	Energy Interoperation, an OASIS specification as per the normative references, CTS is a conforming profile of EI.
EMIX	Energy Market Information Exchange, an OASIS specification used to describe Products and markets for resources, particularly those traded in power grids.
FIX	The Financial Information Exchange Protocol is a vendor-neutral electronic communications protocol for the international real-time exchange of securities transaction information. The FIX Protocol language is comprised of a series of messaging specifications used in trade communications. FIX is maintained by the FIX Trading Community.

Appendix F. Acknowledgments

This work is derived from the specification Common Transactive Services 1.0 , contributed by The Energy Mashup Lab, written by William T. Cox and Toby Considine.

Portions of models and text are derived from The Energy Mashup Lab open-source project EML-CTS and is used with explicit permission under the terms of the Apache 2.0 License for that project.[33]

Deleted: is

Deleted: ,

F.1 Participants

The following individuals were members of this Technical Committee during the creation of this document and their contributions are gratefully acknowledged:

Rolf Bienert, OpenADR Alliance
Toby Considine, University of North Carolina at Chapel Hill
William T. Cox, Individual Member
Pim van der Eijk, Sonnenglanz Consulting
David Holmberg, National Institute for Standards & Technology (NIST)
Elysa Jones, Individual Member
Chuck Thomas, Electric Power Research Institute (EPRI)

Hanno Klein and members of the FIX Trading Community Transactive Resource Working Group

F.2 Special Thanks

The Technical Committee extends a special thanks to Hanno Klein, co-chair of the FIX Global Technical Committee, and Senior Standards advisor at FIXdom. Hanno’s patient explanations of trading semantics and suggestions for approaches that would increase the commonality of financial markets and CTS markets were invaluable. His knowledge of global, open and free standards for trading is unparalleled. Where we missed our mark, it is where we misunderstood his advice.

[33] https://github.com/EnergyMashupLab/eml-cts

2612
Appendix G. Revision History

2613

Revision	Date	Editor	Changes Made
WD01	2/15/2021	Toby Considine	Initial reformatting and conversion of the specification contributed by The Energy Mashup Lab to create a document for committee work.
WD02	3/3/2021	Toby Considine	Added prose definitions of Resource, Product, and Instrument
WD03	4/5/2021	Toby Considine	Simplified introductory material, raised message type to earlier in document. Removed some repetitive material. Revised UML required.
WD04	5/7/2021	Toby Considine David Holmberg William T Cox	Reordered intro material to reduce repetition, Reference Actor Model more consistently, Revise and re-factor Resource/Product/Instrument Add Section 3 to elevate common semantic elements
WD05	5/25/2021	Toby Considine David Holmberg William T Cox	Continues clean-up and condensation of sections 1, 2
WD06	6/7/2021	Toby Considine	Refines Item language into Resource and Products. Explains Message Groups as a conforming descendant of EI Services.
WD07	6/21/2021	Toby Considine William T Cox	Clarified terminology and relationship to implied Service-Oriented Architecture. Structured CTS facets for clearer explanation
WD08	8/5/2021	Toby Considine William T Cox David Holmberg	Clarify and simplify actor facets descriptions, including Tender, Transaction, and Configuration. Reduce redundant and less relevant content.
WD09	9/14/2021	William T Cox Toby Considine David Holmberg	Added Facet descriptions for Position, Market Characteristics, CTS Streams, and drafts of Privacy Consideration, Delivery and Party Registration Facets. Numerous edits for clarity and conciseness.
WD10	10/4/2021	Toby Considine William T Cox David Holmberg	Extended Market Facets. Defined Position and Delivery facets. Made references more consistent. Updated UML model and diagrams.
WD11	10/22/2021	David Holmberg William T Cox Toby Considine	Corrections for clarity. Improved UML diagrams. Flagged requests for comments in Public Review

Revision	Date	Editor	Changes Made
CSD01	10/29/2021	OASIS TC Administration	Content as in WD11, formatted to include OASIS metadata and references to the published specification
WD12	1/10/2022	William T Cox Toby Considine	Simpler edits in response to comments from PR
WD13		William T Cox Toby Considine	Clarification of Resource/Product/Instrument Removal of references to “Architecture” Responses to “Clarity” tagged issues
WD14	2/22/2022	William T Cox Toby Considine	Clarification of front material Section 1/-2 compared to eliminate duplicative definitions Numerous issues resolutions applied as per Jira
WD15	3/20/2020	William T Cox Toby Considine	Clarity, responses to issues from Review
WD16	4/12/2022	William T Cox Toby Considine	Marketplace and Market characteristics responses to issues Expanded Quotes and Tickers Focus on capitalization
WD17	4/25/2022	William T Cox Toby Considine	Updated UML Market Information added OTC Transactions Edits for Clarity
WD18	9/19/2023	Toby Considine	First response to FIX meetings Changed to Market/Market Segment language Reference FIX Tags when known Closings and Crossings added First pass at FIX-conformant Market Data Reports
WD19	10/2024	Toby Considine	Response to Second PR Preparations to work with
WD20-22		Toby Considine	Re-writes while discussing with FIX. Added Negotiations, Tickers, Instrument Data, Market Structure
WD23	6/23/2024	Toby Considine	Post PR03, showing all comments received
WD24	7/7/2024	Toby Considine William T. Cox	Working through public review comments Simplification of Tickers and Market Structure Re-work and simplification of Negotiations
WD25	7/7/2024	Toby Considine	Accepted simpler comments to focus attention to larger issues

Revision	Date	Editor	Changes Made
WD26	7/17/2024	Toby Considine	Moved material on selecting a Market Mechanism and on non-normative illustrations of business interactions to an appendix
WD27	8/12/2024	William T. Cox Toby Considine	Rework of models and exposition for all negotiations & subscriptions (9-12)
WD28	8/27/2024	William T Cox	UML updates and revisions for the entire technical content of the specification.
WD29		William T Cox Toby Considine David Holmberg	Subscriptions, and more consistent delineations of Structure (non-volatile) and Session (volatile) data throughout. Many smaller edits to align earlier parts of document with the specification as it has emerged in later details.
WD30	11/23/2024	Toby Considine	All “simple” FIX comments addressed (wrong variable name, etc.) All other review comments incoporated into document
WD31	1/17/2015	William T Cox Toby Considine David Holmberg	PR Comments processed and applied misc Editorial comments, clarity addressed misc formatting errors
WD32	2/17/202	William T Cox Toby Considine	Detailed analysis and of UML model, tables, and sequence diagrams, which drove minor technical and ediitoral corrections.

Notices

Copyright © OASIS Open 2025. All Rights Reserved.

Deleted: 2024

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website: [https://www.oasis-open.org/policies-guidelines/ipr/].

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

The name "OASIS" is a trademark of OASIS, the owner and developer of this document, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, documents, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark/ for above guidance.