S&P Capital IQ Real-Time Solutions

FeedOS™ Developer's Notice

DUBLIN – Feed Update

Reference n°: 20150507 – 20122 – 26342 – 26452

Effective as of: 01 June 2015*

Action required from users: MANDATORY ACTION

S&P CAPITAL IQ
McGRAW HILL FINANCIAL

* For the actual day when the changes to your custom feed handler take effect, please contact your QuantFEED® project manager.

S&P Capital IQ Real-Time Solutions
FeedOS™ Developer's Notice: DUBLIN – Feed Update
Reference 20150507 – 20122 – 26342 – 26452
May 28, 2015

France
52 Rue de la Victoire
75009 Paris
France
Tel: +33 (0) 1 73 02 32 11

United States
55 Water Street, 44th floor
New York, NY 10041
United States of America
Tel: +1-(212)-438-4346

130 East Randolph
One Prudential Plaza, Suite 2900
Chicago, IL 60601
United States of America
Tel: +1-(312)-233-7129

United Kingdom
20 Canada Square
Canary Wharf
London E14 5LH
United Kingdom
Tel: +44 (0) 203 107 1676

Singapore
12 Marina Boulevard
#23-01 Marina Bay
Financial Centre Tower 3
Singapore 018982
Tel: +65 6530 6546

www.spcapitaliq.com

No content (including ratings, credit-related analyses and data, valuations, model, software or other application or output therefrom) or any part thereof (Content) may be modified, reverse engineered, reproduced or distributed in any form by any means, or stored in a database or retrieval system, without the prior written permission of Standard & Poor's Financial Services LLC or its affiliates (collectively, S&P). The Content shall not be used for any unlawful or unauthorized purposes. S&P and any third-party providers, as well as their directors, officers, shareholders, employees or agents (collectively S&P Parties) do not guarantee the accuracy, completeness, timeliness or availability of the Content. S&P Parties are not responsible for any errors or omissions (negligent or otherwise), regardless of the cause, for the results obtained from the use of the Content, or for the security or maintenance of any data input by the user. The Content is provided on an "as is" basis. S&P PARTIES DISCLAIM ANY AND ALL EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE, FREEDOM FROM BUGS, SOFTWARE ERRORS OR DEFECTS, THAT THE CONTENT'S FUNCTIONING WILL BE UNINTERRUPTED OR THAT THE CONTENT WILL OPERATE WITH ANY SOFTWARE OR HARDWARE CONFIGURATION. In no event shall S&P Parties be liable to any party for any direct, indirect, incidental, exemplary, compensatory, punitive, special or consequential damages, costs, expenses, legal fees, or losses (including, without limitation, lost income or lost profits and opportunity costs or losses caused by negligence) in connection with any use of the Content even if advised of the possibility of such damages.

Credit-related and other analyses, including ratings, and statements in the Content are statements of opinion as of the date they are expressed and not statements of fact or recommendations to purchase, hold, or sell any securities or to make any investment decisions. S&P assumes no obligation to update the Content following publication in any form or format. The Content should not be relied on and is not a substitute for the skill, judgment and experience of the user, its management, employees, advisors and/or clients when making investment and other business decisions. S&P's opinions and analyses do not address the suitability of any security. S&P does not act as a fiduciary or an investment advisor except where registered as such. While S&P has obtained information from sources it believes to be reliable, S&P does not perform an audit and undertakes no duty of due diligence or independent verification of any information it receives.

S&P keeps certain activities of its business units separate from each other in order to preserve the independence and objectivity of their respective activities. As a result, certain business units of S&P may have information that is not available to other S&P business units. S&P has established policies and procedures to maintain the confidentiality of certain non-public information received in connection with each analytical process.

UPDATE OF THE DUBLIN MARKET DATA STREAM

To reflect the changes caused by the migration of the DUBLIN market data stream to the XETRA ULTRA PLUS protocol, S&P Capital IQ Real-Time Solutions has decided to enhance the content of FeedOS™.

This developer's notice contains late-breaking information about the implementation of this modification in your applications, which may not be included otherwise in the published documentation. The topics this notice covers include:

1. Update Summary

Table 1 Current update summary

Notice Reference	20150507 – 20122 – 26342 – 26452
Exchanges	DUBLIN
Concerned MICs	XDUB
Internal Source ID	23, 24, 25, 26, 27, 51, 87, 93, 94
Effective Date	**2015-06-01***
Impact	• Update of the Referential Tags • Update of the Quotation Tags • Update of the Quotation Context Tags • Update of the Level1 Market Data Kinematics
Action required	**MANDATORY ACTION** - see sections: • 2.1.14. SecurityType • 2.1.15. CFICode • 2.2.5. TradingStatus • 2.4. Removal of the Off Book Trades from the Level1 Market Data • 2.5. Addition of the Number of Orders to the BBO of the Level1 Market Data • 2.6. MBL and MBO Data.

2. FeedOS Technical Implementation

Effective Monday, **June 01**[*] **2015,** S&P Capital IQ Real-Time Solutions enhances the referential and quotation data, and updates the Level1 Market Data Kinematics to accommodate the information disseminated on the DUBLIN market data stream, as described below:

- 2.1. Changes to the Referential Data
- 2.2. Changes to the Quotation Data
- 2.3. Changes to the Quotation Context Data
- 2.4. Removal of the Off Book Trades from the Level1 Market Data
- 2.5. Addition of the Number of Orders to the BBO of the Level1 Market Data
- 2.6. MBL and MBO Data
- 2.7. Microsecond Timestamp Precision on the Level1 Market Data.

2.1. Changes to the Referential Data

S&P Capital IQ Real-Time Solutions **introduces** the referential tags below to accommodate the information disseminated on the DUBLIN market data stream:

Table 2 Referential tags added on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`Issuer`	106	String
`StdMaturity`	200	String
`CouponPaymentDate`	224	UInt32
`RoundLot`	561	Float64
`MinTradeVol`	562	Float64
`SecuritySubType`	762	String
`MaturityYear`	9512	Int16
`MaturityMonth`	9513	Uint8
`MaturityDay`	9514	Uint8
`CCP_Eligible`	9552	Book
`MARKET_XETRA_ISIX`	11101	UInt32
`MARKET_XETRA_OptimalGatewayLocation`	11102	String
`MARKET_XETRA_CCP_Eligible`	11103	Bool

[*] This is the proposed day for the update of the standard version of the feed handler. For dedicated feed handlers, the date and Source IDs may differ. For the actual day when the changes to your custom feed handler will take effect, please contact your FeedOS™ project manager.

Moreover, S&P Capital IQ Real-Time Solutions **updates** the referential tags below:

Table 3 Referential tags disseminating updated values on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`SecurityType`	167	String
`CFICode`	461	String

S&P Capital IQ Real-Time Solutions also **removes** the referential tags below:

Table 4 Referential tags no longer disseminated on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`InternalAggregationId`	9404	UInt16
`MARKET_XETRA_SegmentCode`	11100	String

2.1.1. Issuer

The values of the referential tag **Issuer** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to detail the issuer of a security.

FeedOS implementation of the tag `Issuer` is described in the table below:

Table 5 Issuer – technical implementation in FeedOS

Component	Value	Description
Tag Name	`Issuer`	FeedOS tag name.
Numeric ID	106	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the issuer of a security.

2.1.2. StdMaturity

The values of the referential tag **StdMaturity** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the standard maturity of a security.

FeedOS implementation of the tag `StdMaturity` is described in the table below:

Table 6 StdMaturity – technical implementation in FeedOS

Component	Value	Description
Tag Name	`StdMaturity`	FeedOS tag name.
Numeric ID	200	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	`[YYYYMM]`	Year-Month Format
Possible values	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the standard maturity of a security.

2.1.3. CouponPaymentDate

The values of the referential tag **CouponPaymentDate** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the date when the interest for a bond, note or fixed income security is to be paid.

FeedOS implementation of the tag `CouponPaymentDate` is described in the table below:

Table 7 CouponPaymentDate – technical implementation in FeedOS

Component	Value	Description
Tag Name	`CouponPaymentDate`	FeedOS tag name.
Numeric ID	`224`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`UInt32`	UInt32 data type.
Format / Possible Value	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the date when the interest for a bond, note or fixed income security is to be paid.

2.1.4. RoundLot

The values of the referential tag **RoundLot** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the smallest order that can be placed.

FeedOS implementation of the tag `RoundLot` is described in the table below:

Table 8 RoundLot – technical implementation in FeedOS

Component	Value	Description
Tag Name	`RoundLot`	FeedOS tag name.
Numeric ID	`561`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the smallest order that can be placed.

2.1.5. MinTradeVol

The values of the referential tag **MinTradeVol** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the minimum traded volume.

FeedOS implementation of the tag `MinTradeVol` is detailed in the table below:

Table 9 MinTradeVol – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MinTradeVol`	FeedOS tag name.
Numeric ID	`562`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the minimum traded volume.

2.1.6. SecuritySubType

The values of the referential tag **SecuritySubType** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify additional details about the securities associated with the market CFI Codes.

FeedOS implementation of the values currently available for the tag `SecuritySubType` is described in the table below:

Table 10 SecuritySubType – technical implementation in FeedOS

Component	Value	Description
Tag Name	`SecuritySubType`	FeedOS tag name.
Numeric ID	`762`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the securities associated with the market CFI Codes.

2.1.7. MaturityYear

The values of the referential tag **MaturityYear** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the year on which the principal is required to be repaid.

FeedOS implementation of the tag `MaturityYear` is described in the table below:

Table 11 MaturityYear – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MaturityYear`	FeedOS tag name.
Numeric ID	`9512`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Int16`	Int16 data type.
Format / Possible Value	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the year on which the principal is required to be repaid.

2.1.8. MaturityMonth

The values of the referential tag **MaturityMonth** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the month on which the principal is required to be repaid.

FeedOS implementation of the tag `MaturityMonth` is described in the table below:

Table 12 MaturityMonth – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MaturityMonth`	FeedOS tag name.
Numeric ID	`9513`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`UInt8`	UInt8 data type.
Format / Possible Value	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the month on which the principal is required to be repaid.

2.1.9. MaturityDay

The values of the referential tag **MaturityDay** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the day on which the principal is required to be repaid.

FeedOS implementation of the tag `MaturityDay` is described in the table below:

Table 13 MaturityDay – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MaturityDay`	FeedOS tag name.
Numeric ID	`9514`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`UInt8`	UInt8 data type.
Format / Possible Value	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying the day on which the principal is required to be repaid.

2.1.10. CCP_Eligible

The values of the referential tag **CCP_Eligible** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify whether an instrument is cleared via the CCP or not.

FeedOS implementation of the values currently available for the tag `CCP_Eligible` is described in the following table:

Table 14 CCP_Eligible – technical implementation in FeedOS

Component	Value	Description
Tag Name	`CCP_Eligible`	FeedOS tag name.
Numeric ID	`9552`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.

Table 14 CCP_Eligible – technical implementation in FeedOS (Continued)

Component	Value	Description
Type	`Bool`	Bool data type.
Format	*[Exchange Specific Value]*	An ***exchange specific value***, detailing whether an instrument is cleared via the CCP.
Possible Values	`True`	CCP eligibility and post trade anonymity.
	`False`	Default value, not sent.

2.1.11. MARKET_XETRA_ISIX

The values of the referential tag **MARKET_XETRA_ISIX** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to uniquely identify an instrument across the system.

FeedOS implementation of the tag `MARKET_XETRA_ISIX` is described in the table below:

Table 15 MARKET_XETRA_ISIX – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_ISIX`	FeedOS tag name.
Numeric ID	`11101`	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`UInt32`	UInt32 data type.
Format / Possible Values	*[Exchange Specific Value]*	An ***exchange specific value***, uniquely identifying an instrument across the system.

2.1.12. MARKET_XETRA_OptimalGatewayLocation

The values of the referential tag **MARKET_XETRA_OptimalGatewayLocation** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to identify the optimal performance gateway location for trading the instrument.

FeedOS implementation of the tag `MARKET_XETRA_OptimalGatewayLocation` is described in the table below:

Table 16 MARKET_XETRA_OptimalGatewayLocation – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_OptimalGatewayLocation`	FeedOS tag name.
Numeric ID	`11102`	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*[Exchange Specific Value]*	An ***exchange specific value***, identifying the optimal performance gateway location for trading the instrument.

2.1.13. MARKET_XETRA_CCP_Eligible

The values of the referential tag **MARKET_XETRA_CCP_Eligible** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specific whether an instrument is cleared via the CCP or not.

FeedOS implementation of the tag `MARKET_XETRA_CCP_Eligible` is described in the table below:

Table 17 MARKET_XETRA_CCP_Eligible – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_CCP_Eligible`	FeedOS tag name.
Numeric ID	`11103`	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Bool`	Bool data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing whether an instrument is cleared via the CCP or not.
Possible Values	`True`	CCP eligibility and post trade anonymity.
	`False`	Default value, not sent.

2.1.14. SecurityType

The values of the referential tag **Security Type** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the type of security.

FeedOS implementation of the tag `SecurityType` is described in the table below (existing values are in `black`, newly added values are in `green`, removed values are in ~~`crossed out red`~~):

Table 18 SecurityType – technical implementation in FeedOS

Component	Value	Description
Tag Name	`SecurityType`	FeedOS tag name.
Numeric ID	`167`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the type of security.
Possible Values	`CS`	Common Stock
	`GO`	General Obligation
	~~`INDEX`~~	Index
	~~`NONE`~~	None
	~~`WAR`~~	Warrant

2.1.15. CFICode

The values of the referential tag **CFI Code** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Referential* to specify the standardized identification code of an instrument.

FeedOS implementation of the tag `CFICode` is described in the table below (existing values are in `black`, newly added values are in `green`, removed values are in ~~`crossed out red`~~):

Table 19 CFICode – technical implementation in FeedOS

Component	Value	Description
Tag Name	`CFICode`	FeedOS tag name.
Numeric ID	`461`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the standardized identification code of an instrument.
Possible Values	~~`DBXXXX`~~	Debts - Bonds
	`DBZXXX`	Debts - Bonds - Zero Rate
	~~`EUXXXX`~~	Equities - Units
	`EXXXXX`	Equities
	~~`MRIXXX`~~	Others - Referential Instruments - Indices
	~~`RWXXXX`~~	Rights - Warrants

The example below shows the possible combinations of `SecurityTypes` and `CFICodes`, before and after the migration day (please note that additional combinations may be available, as the exchange could introduce new instruments):

```
BEFORE 2015-06-01                           AFTER 2015-06-01
{ XDUB INDEX MRIXXX }                       { XDUB CS    EXXXXX }
{ XDUB NONE  DBXXXX }                       { XDUB GO    DBZXXX }
{ XDUB NONE  EUXXXX }
{ XDUB NONE  EXXXXX }
{ XDUB WAR   RWXXXX }
```

Referential Data Sample

Below are several examples showing the current implementation of the newly added (in `green`), updated (in `blue`) and removed (in ~~`crossed out red`~~) referential tags:

- Equities
- Bonds.

Equities

```
BEFORE 2015-06-01
instr # 125/1048 = 262145048
    PriceCurrency               string{EUR}
    Symbol                      string{BIR}
    Description                 string{BK OF IRELD CAP.ST.EO-,05}
    SecurityType                string{NONE}
    FOSMarketId                 XDUB
    CFICode                     string{EXXXXX}
    SecurityGroup               string{ISE1}
    InternalCreationDate        Timestamp{2014-06-09 07:02:59:557}
    InternalModificationDate    Timestamp{2015-05-28 05:15:30:857}
    InternalSourceId            uint16{87}
    ~~InternalAggregationId~~       ~~uint16{87}~~
    InternalEntitlementId       int32{1045}
    LocalCodeStr                string{IE0030606259}
    ISIN                        string{IE0030606259}
    WertpapierKennNummer        string{853701}
    PriceIncrement_dynamic_TableId      uint32{3342436}
    OperatingMIC                string{XDUB}
    ~~MARKET_XETRA_SegmentCode~~    ~~string{None}~~


AFTER 2015-06-01
instr # 125/8291 = 262152291
    PriceCurrency               string{EUR}
    Symbol                      string{BIR}
    Description                 string{BANK OF IRELAND}
    SecurityType                string{CS}
    FOSMarketId                 XDUB
    CFICode                     string{EXXXXX}
    RoundLot                    float64{1}
    MinTradeVol                 float64{1}
    SecuritySubType             string{EQU}
    SecurityGroup               string{ISE1}
    InternalCreationDate        Timestamp{2015-04-28 13:47:39:958}
    InternalModificationDate    Timestamp{2015-04-28 13:47:59:791}
    InternalSourceId            uint16{158}
    InternalEntitlementId       int32{999}
    LocalCodeStr                string{IE0030606259}
    ISIN                        string{IE0030606259}
    WertpapierKennNummer        string{853701}
    PriceIncrement_dynamic_TableId      uint32{10354788}
    OperatingMIC                string{XDUB}
    CCP_Eligible                bool{True}
    MARKET_XETRA_ISIX           uint32{4534}
    MARKET_XETRA_OptimalGatewayLocation string{0001}
    MARKET_XETRA_CCP_Eligible   bool{True}
```

Bonds

```
BEFORE 2015-06-01
instr # 125/2690 = 262146690
    PriceCurrency               string{EUR}
    Symbol                      string{1I4P}
    Description                 string{ABBEY NATL. NOTES 18 S500}
    SecurityType                string{WAR}
    FOSMarketId                 XDUB
    CFICode                     string{RWXXXX}
    SecurityGroup               string{ISED}
    InternalCreationDate        Timestamp{2014-06-09 05:15:03:773}
    InternalModificationDate    Timestamp{2015-05-28 05:15:32:830}
    InternalSourceId            uint16{87}
    ~~InternalAggregationId~~       ~~uint16{87}~~
    InternalEntitlementId       int32{1045}
    LocalCodeStr                string{XS0810173285}
    ISIN                        string{XS0810173285}
    PriceIncrement_static       float64{0.001}
    WertpapierKennNummer        string{A1RTBN}
    OperatingMIC                string{XDUB}
    ~~MARKET_XETRA_SegmentCode~~    ~~string{None}~~


AFTER 2015-06-01
instr # 125/9402 = 262153402
    PriceCurrency               string{EUR}
    Symbol                      string{1I4P}
    Issuer                      string{0001}
    Description                 string{ABBZERO%05.10.2018}
    SecurityType                string{GO}
    StdMaturity                 string{201810}
    FOSMarketId                 XDUB
    CouponPaymentDate           uint32{20151005}
    CFICode                     string{DBZXXX}
    RoundLot                    float64{100000}
    MinTradeVol                 float64{100000}
    SecuritySubType             string{BON}
    SecurityGroup               string{ISED}
    InternalCreationDate        Timestamp{2015-04-28 09:17:25:911}
    InternalModificationDate    Timestamp{2015-04-28 09:17:45:387}
    InternalSourceId            uint16{158}
    InternalEntitlementId       int32{1209}
    LocalCodeStr                string{XS0810173285}
    ISIN                        string{XS0810173285}
    PriceIncrement_static       float64{0.001}
    MaturityYear                uint16{2018}
    MaturityMonth               uint8{10}
    MaturityDay                 uint8{5}
    WertpapierKennNummer        string{A1RTBN}
    OperatingMIC                string{XDUB}
    CCP_Eligible                bool{False}
    MARKET_XETRA_ISIX           uint32{483}
    MARKET_XETRA_OptimalGatewayLocation string{0001}
    MARKET_XETRA_CCP_Eligible   bool{False}
```

2.2. Changes to the Quotation Data

S&P Capital IQ Real-Time Solutions **introduces** the quotation tags below to accommodate the information disseminated on the DUBLIN market data stream:

Table 20 Quotation tags added on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`LastAuctionPrice`	9146	Float64
`LastAuctionVolume`	9147	Float64
`PriceActivityMarketTimestamp`	9309	Timestamp
`MARKET_XETRA_ULTRA_PLUS_InstrumentStatus`	14480	Float64

Moreover, S&P Capital IQ Real-Time Solutions **updates** the quotation tags below:

Table 21 Quotation tags disseminating updated values on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`TradingStatus`	9100	Enum

S&P Capital IQ Real-Time Solutions also **removes** the quotation tags below:

Table 22 Quotation tags no longer disseminated on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`LastOffBookTradePrice`	9110	Float64
`LastOffBookTradeQty`	9111	Float64
`LastOffBookTradeTimestamp`	9112	Timestamp
`DailyTotalOffBookVolumeTraded`	9148	Float64
`DailyTotalOffBookAssetTraded`	9149	Float64
`MARKET_CEF_LastTradeTradingPhase`	14900	Char

2.2.1. LastAuctionPrice

The values of the quotation tag **LastAuctionPrice** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Other Values* to detail the last price:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `LastAuctionPrice` is described in the following table:

Table 23 LastAuctionPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`LastAuctionPrice`	FeedOS tag name.
Numeric ID	`9146`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the last auction price.

2.2.2. LastAuctionVolume

The values of the quotation tag **LastAuctionVolume** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Other Values* to detail the last volume:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `LastAuctionVolume` is described in the following table:

Table 24 LastAuctionVolume – technical implementation in FeedOS

Component	Value	Description
Tag Name	`LastAuctionVolume`	FeedOS tag name.
Numeric ID	`9147`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the last auction volume.

2.2.3. PriceActivityMarketTimestamp

The values of the quotation tag **PriceActivityMarketTimestamp** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the time of the last change of a book or trade, in terms if Last Price, Bid or Ask:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `PriceActivityMarketTimestamp` is described below:

Table 25 PriceActivityMarketTimestamp – technical implementation in FeedOS

Component	Value	Description
Tag Name	`PriceActivityMarketTimestamp`	FeedOS tag name.
Numeric ID	`9309`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Timestamp`	Timestamp data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, indicating the time of the last change of a book or trade, in terms of Last Price, Bid or Ask.

2.2.4. MARKET_XETRA_ULTRA_PLUS_InstrumentStatus

Each time a change of the instrument status occurs, the values of the quotation tag **MARKET_XETRA_ULTRA_PLUS_InstrumentStatus** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Other Values*:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++

- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag **MARKET_XETRA_ULTRA_PLUS_InstrumentStatus** is described in the table below:

Table 26 MARKET_XETRA_ULTRA_PLUS_InstrumentStatus – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_ULTRA_PLUS_InstrumentStatus`	FeedOS tag name.
Numeric ID	`14480`	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. It is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, as described below, concerning the status of the instrument.
Possible Values	0	Start
	1	Pre Trading
	2	Pre-call
	3	Crossing Period
	4	Closing Crossing Period
	5	Opening Auction Call
	6	Intra Day Auction Call
	7	Closing Auction Call
	8	End Auction Call
	9	Auction Call
	10	Opening Auction IPO Call
	11	Opening Auction IPO Freeze
	12	Intra Day Auction IPO Call
	13	Intra Day Auction IPO Freeze
	14	IPO
	15	Quote Driven IPO Freeze
	16	Opening Auction Pre-Orderbook Balancing
	17	Intra Day Auction Pre-Orderbook Balancing
	18	Closing Auction Pre-Orderbook Balancing
	19	End-of-day Auction Pre-Orderbook Balancing
	20	Pre-Orderbook Balancing of Quote Driver Auction
	21	Opening Auction Orderbook Balancing
	22	Intra Day Auction Orderbook Balancing
	23	Closing Auction Orderbook Balancing
	24	End-of-day Auction Orderbook Balancing
	25	Orderbook Balancing
	26	Continuous Trading
	27	In Between Auctions
	28	Post Trading
	29	End of Trading

Table 26 MARKET_XETRA_ULTRA_PLUS_InstrumentStatus – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	30	Halt
	31	Suspend
	32	Volatility Interruption
	35	Add
	36	Delete
	38	Call Unfreeze
	39	Continuous Auction Pre-Call
	40	Continuous Auction Call
	41	Continuous Auction Freeze
	51	Knocked Out
	52	Knocked Out / Revoked
	53	Midpoint Book Freeze
	54	Midpoint Book Unfreeze

2.2.5. TradingStatus

Each time a modification of the trading status occurs, the values of the quotation tag **TradingStatus** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Other Values*:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `TradingStatus` is described in the following table (removed values are in ~~`crossed out red`~~):

Table 27 TradingStatus – technical implementation in FeedOS

Component	Value	Description
Tag Name	`TradingStatus`	FeedOS tag name.
Numeric ID	9100	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Enum`	Enum data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the characteristics of the trading status.
Possible Values	2	Trading Halt
	5	Price Indication
	15	New Price Indication
	17	Ready to Trade
	18	Not Available for Trading
	21	Pre-Open
	~~23~~	Fast Market

Quotation Data Sample

Below is an example showing the current implementation of the newly added (in green), updated (in blue) and removed (in ~~crossed out red~~) quotation tags:

```
BEFORE 2015-06-01
InstrumentStatusL1
-- 125/1066
        BID: 1.1        5000
        ASK: 1.105      5000
        LastPrice                       float64{1.105}
        LastTradeQty                    float64{5000}
        DailyHighPrice                  float64{1.105}
        DailyLowPrice                   float64{1.104}
        DailyTotalVolumeTraded          float64{523541}
        DailyTotalAssetTraded           float64{578509.718}
        LastTradePrice                  float64{1.105}
        LastTradeTimestamp              Timestamp{2015-04-29 09:39:35}
        InternalDailyOpenTimestamp      Timestamp{2015-04-29 07:00:10:030}
        InternalDailyCloseTimestamp     Timestamp{2015-04-28 15:31:17:084}
        InternalDailyHighTimestamp      Timestamp{2015-04-29 08:05:39:105}
        InternalDailyLowTimestamp       Timestamp{2015-04-29 08:09:11:370}
        InternalPriceActivityTimestamp  Timestamp{2015-04-29 09:39:35:101}
        TradingStatus                   17=ReadyToTrade
        ~~LastOffBookTradePrice~~       ~~float64{1.105}~~
        ~~LastOffBookTradeQty~~         ~~float64{500000}~~
        ~~LastOffBookTradeTimestamp~~   ~~Timestamp{2015-04-29 08:26:28:530}~~
        DailyOpeningPrice               float64{1.105}
        PreviousDailyTotalVolumeTraded  float64{67361}
        PreviousDailyTotalAssetTraded   float64{74438.104}
        PreviousDailyClosingPrice       float64{1.105}
        PreviousBusinessDay             Timestamp{2015-04-28}
        CurrentBusinessDay              Timestamp{2015-04-29}
        ~~DailyTotalOffBookVolumeTraded~~ ~~float64{1330337}~~
        ~~DailyTotalOffBookAssetTraded~~ ~~float64{1470022.385}~~
        LastAuctionImbalanceSide        char{B}
        LastAuctionImbalanceVolume      float64{5748}
        InternalDailyClosingPriceType   char{a}
        PreviousInternalDailyClosingPriceType   char{a}
        InternalLastAuctionTimestamp    Timestamp{2015-04-28 15:30:46:090}
        ~~MARKET_CEF_LastTradeTradingPhase~~    ~~char{C}~~
```

```
AFTER 2015-06-01
InstrumentStatusL1
-- 125/8300
        BID: 1.1        5000    @1
        ASK: 1.106      4347    @1
        LastPrice                       float64{1.105}
        LastTradeQty                    float64{10000}
        DailyHighPrice                  float64{1.105}
        DailyLowPrice                   float64{1.104}
        DailyTotalVolumeTraded          float64{431567}
        DailyTotalAssetTraded           float64{476880.422}
        LastTradePrice                  float64{1.105}
        LastTradeTimestamp              Timestamp{2015-04-29 09:24:33:489}
        InternalDailyOpenTimestamp      Timestamp{2015-04-29 07:00:10:019}
        InternalDailyCloseTimestamp     Timestamp{2015-04-28 15:31:17:072}
        InternalDailyHighTimestamp      Timestamp{2015-04-28 15:15:35:324}
        InternalDailyLowTimestamp       Timestamp{2015-04-28 14:07:00:792}
        InternalPriceActivityTimestamp  Timestamp{2015-04-29 09:24:50:642}
        TradingStatus                   17=ReadyToTrade
        DailyOpeningPrice               float64{1.105}
        PreviousDailyTotalVolumeTraded  float64{67361}
        PreviousDailyTotalAssetTraded   float64{7452.528}
        PreviousDailyClosingPrice       float64{1.105}
        PreviousBusinessDay             Timestamp{2015-04-28}
        CurrentBusinessDay              Timestamp{2015-04-29}
        LastAuctionPrice                float64{1.105}
        LastAuctionVolume               float64{252}
        LastAuctionImbalanceSide        char{B}
        LastAuctionImbalanceVolume      float64{5748}
        PreviousInternalDailyClosingPriceType   char{a}
        InternalLastAuctionTimestamp    Timestamp{2015-04-28 15:30:15:127}
        PriceActivityMarketTimestamp    Timestamp{2015-04-29 09:24:50:642}
        MARKET_XETRA_ULTRA_PLUS_InstrumentStatus        float64{26}
```

2.3. Changes to the Quotation Context Data

S&P Capital IQ Real-Time Solutions **introduces** the quotation context tags below to accommodate the information disseminated on the DUBLIN market data stream:

Table 28 Quotation context tags added on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`MARKET_XETRA_ULTRA_PLUS_TradeType`	15900	String
`MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator`	15901	Char

S&P Capital IQ Real-Time Solutions also **removes** the quotation context tags below:

Table 29 Quotation context tags no longer disseminated on the DUBLIN market data stream

Tag Name	Numeric ID	Type
`MARKET_CEF_IndexTypeIndicator`	15150	String
`MARKET_CEF_LastAuctionQty`	15151	Float64
`MARKET_CEF_TradeTypeIndicator`	15400	String

2.3.1. MARKET_XETRA_ULTRA_PLUS_TradeType

Each time a trade occurs, the values of the quotation tag **MARKET_XETRA_ULTRA_PLUS_TradeType** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Context* to detail the trade type:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `XETRA_Ultra_Plus_Trade_Type` is described in the table below:

Table 30 XETRA_Ultra_Plus_Trade_Type – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_ULTRA_PLUS_TradeType`	FeedOS tag name.
Numeric ID	`15900`	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. It is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, as described below, concerning the characteristics of the trade type.
Possible Values	`4`	Last traded price (it indicates the normal trade; by default, not sent).
	`9`	Price from the subscription period
	`10`	BEST price
	`11`	Midpoint order trade
	`25`	Price determined with Bundesbank participation

2.3.2. MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator

Each time a trade occurs, the values of the quotation tag **MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator** conveyed on the DUBLIN market data stream are disseminated via FeedOS data stream in *Context* to detail the type of trade:

- in the callback carrying the Level1 event `notif_TradeEventExt()`, for C++
- in the event handler `TradeEventExtEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

FeedOS implementation of the tag `MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator` is described in the table below:

Table 31 MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_XETRA_ULTRA_PLUS_TradeType Indicator`	FeedOS tag name.
Numeric ID	`15901`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Char`	Char data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the type of trade.

Table 31 MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	A	Auction
	C	Continuous Trading
	E	End-of-Day Auction
	F	Closing Auction
	L	Liquidity Interruption
	M	Mini Auction
	O	Opening Auction
	V	Volatility / Interruption in Continuous Trading

Quotation Context Data Sample

Below is an example showing the current implementation of the newly added (in `green`) and removed (in ~~`crossed out red`~~) quotation tags:

```
BEFORE 2015-06-01
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

VU   05:30:00:192    262152291    TradingStatus=15
TE   06:50:00:051    262152291    *       *       0.4     25      0.35    52000
~~MARKET_CEF_LastAuctionQty=float64{25}~~
VU   06:50:00:051    262152291    LastAuctionPrice=0.35   LastAuctionVolume=25
VU   06:50:00:051    262152291    TradingStatus=21
TE   06:50:01:200    262152291    *       *       *       *       0.35    64000
TE   06:50:01:658    262152291    *       *       *       *       *       *
~~MARKET_CEF_LastAuctionQty=float64{44010}~~
VU   06:50:01:658    262152291    LastAuctionVolume=44010
[...]
TE   12:14:55:976    262152291    0.36    51108   *       *       *       *
TE   12:15:16:905    262152291    0.36    2750000 *       *       *       *       f
~~MARKET_CEF_TradeTypeIndicator=O~~
TE   12:15:16:959    262152291    *       *       0.36    204496  *       *
```

```
AFTER 2015-06-01
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE   04:00:00:408.035    262152291    *       *       !       0       !       0
VU   05:30:00:147.410    262152291    MARKET_XETRA_ULTRA_PLUS_InstrumentStatus=1
TradingStatus=15
TE   06:50:00:037.669    262152291    *       *       0.4     25@1    0.35    52000@1
VU   06:50:00:037.669    262152291    MARKET_XETRA_ULTRA_PLUS_InstrumentStatus=5
LastAuctionPrice=0.35   LastAuctionVo lume=25    TradingStatus=21
TE   06:50:01:187.203    262152291    *       *       *       *       0.35    64000@2
VU   06:50:01:651.318    262152291    LastAuctionVolume=44010
[...]
TE   12:14:55:964.113    262152291    0.36    51108   *       *       *       *
MARKET_XETRA_ULTRA_PLUS_TradeType=string{11},
MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator=char{C}
TE   12:15:16:947.023    262152291    *       *       0.36    204496@1        *       *
```

2.4. Removal of the Off Book Trades from the Level1 Market Data

In the Level1 Market Data Kinematics **before 2015-06-01**, the Off Book Trades were flagged in the Level1 Market Data, as shown in the example below:

```
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE   12:14:55:976    262152291    0.36    51108   *       *       *       *
TE   12:15:16:905    262152291    0.36    2750000 *       *       *       *       f
MARKET_CEF_TradeTypeIndicator=O
TE   12:15:16:959    262152291    *       *       0.36    204496  *       *
```

In the Level1 Market Data Kinematics **after 2015-06-01**, the Off Book Trades are no longer flagged, as shown in the example below:

```
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE   12:14:55:964.113    262152291    0.36    51108   *       *       *       *
MARKET_XETRA_ULTRA_PLUS_TradeTypeIndicator=char{C}
TE   12:15:16:947.023    262152291    *       *       0.36    204496@1        *       *
```

2.5. Addition of the Number of Orders to the BBO of the Level1 Market Data

In the Level1 Market Data Kinematics **before 2015-06-01**, the BBO did not display the number of orders, as shown in the example below:

```
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

VU   05:30:00:192    262152291    TradingStatus=15
TE   06:50:00:051    262152291    *       *       0.4     25      0.35    52000
MARKET_CEF_LastAuctionQty=float64{25}
VU   06:50:00:051    262152291    LastAuctionPrice=0.35   LastAuctionVolume=25
VU   06:50:00:051    262152291    TradingStatus=21
TE   06:50:01:200    262152291    *       *       *       *       0.35    64000
TE   06:50:01:658    262152291    *       *       *       *       *       *
MARKET_CEF_LastAuctionQty=float64{44010}
```

In the Level1 Market Data Kinematics **after 2015-06-01**, the BBO will display the number of orders, as shown in the example below:

```
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE   04:00:00:408.035    262152291    *       *       !       0       !       0
VU   05:30:00:147.410    262152291    MARKET_XETRA_ULTRA_PLUS_InstrumentStatus=1
TradingStatus=15
TE   06:50:00:037.669    262152291    *       *       0.4     25@1    0.35    52000@1
VU   06:50:00:037.669    262152291    MARKET_XETRA_ULTRA_PLUS_InstrumentStatus=5
LastAuctionPrice=0.35   LastAuctionVo lume=25    TradingStatus=21
TE   06:50:01:187.203    262152291    *       *       *       *       0.35    64000@2
VU   06:50:01:651.318    262152291    LastAuctionVolume=44010
```

2.6. MBL and MBO Data[*]

Effective **2015-06-01**, the MBL book has a 10-level depth. There is no MBO.

[*] The MBL and MBO data may not be included by default in your Level1 data subscription, but sold separately. Depending on your contract, additional terms, conditions and fees may apply. For more details about the subscription options, please contact S&P Capital IQ Real-Time Solutions.

2.7. Microsecond Timestamp Precision on the Level1 Market Data

Effective **2015-06-01**, the server timestamps will display microsecond units on the Level1 Market Data, as shown in the example below (highlighted in green):

```
“TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"

TE   19:55:07:508.521   262152291    *       *       *       *       1.27   700@2
TE   20:00:48:238.168   262152291    *       *       *       *       1.22   100@1
TE   20:00:48:240.254   262152291    *       *       *       *       1.31   100@1
```

3. Finding the Latest Information

For the latest documentation and product updates, additional support and training, please contact our support services:

- E-mail: rts-support@spcapitaliq.com
- Web: https://support.quanthouse.com.