S&P Capital IQ Real-Time Solutions

# FeedOS™ Feed Description

## OSLO MIT

Reference n°: 20150402 – 23726 – 24677

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: OSLO MIT
Reference 20150402 – 23726 – 24677
April 03, 2015

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

**US Offices**
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

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

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

www.capitaliq.com

Copyright © 2015 by Standard & Poor's Financial Services LLC, a part of McGraw Hill Financial.

All rights reserved. S&P CAPITAL IQ is a trademark of Standard & Poor's Financial Services LLC. STANDARD & POOR'S, S&P, GLOBAL CREDIT PORTAL and RATINGSDIRECT are registered trademarks of Standard & Poor's Financial Services LLC.

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.

# TABLE OF CONTENTS


**FeedOS™ OSLO MIT Feed Description** ..... 1

1. Referential Data ..... 1
   - 1.1. Available Markets and Branches ..... 1
     - 1.1.1. Markets ..... 2
     - 1.1.2. Branches ..... 2
   - 1.2. Types of Instruments ..... 3
     - 1.2.1. Equities ..... 3
     - 1.2.2. Rights ..... 4
     - 1.2.3. Warrants ..... 5
   - 1.3. Specific Referential Tags ..... 5
     - 1.3.1. SecurityStatus ..... 5
     - 1.3.2. OperatingMIC and SegmentMIC ..... 6
     - 1.3.3. DynamicVariationRange ..... 6
     - 1.3.4. StaticVariationRange ..... 7
     - 1.3.5. MARKET_LSE_NormalMarketSize ..... 7
     - 1.3.6. MARKET_LSE_SegmentCode ..... 8

2. Quotation Data ..... 9
   - 2.1. Quotation Values ..... 10
   - 2.2. TradingStatus ..... 10
   - 2.3. Specific Quotation Tags ..... 11
     - 2.3.1. Trade Conditions ..... 11
       - 2.3.1.1. TradeID ..... 11
       - 2.3.1.2. MMTFlagsV2 ..... 12
       - 2.3.1.3. Aggressor Side ..... 13
       - 2.3.1.4. MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator ..... 14
       - 2.3.1.5. MARKET_LSE_MIT_AuctionTypeIndicator ..... 15
     - 2.3.2. Other Values ..... 15
       - 2.3.2.1. InternalDailyClosingPriceType ..... 15
       - 2.3.2.2. MARKET_TURQUOISE_DarkBookTradingStatus ..... 16
       - 2.3.2.3. MARKET_TURQUOISE_OffBookReportingTradingStatus ..... 17
       - 2.3.2.4. MARKET_LSE_MIT_TradingStatusDetails ..... 17
       - 2.3.2.5. MARKET_LSE_MIT_HaltReason ..... 18
       - 2.3.2.6. MARKET_LSE_MIT_TotalAuctionVolume ..... 19
   - 2.4. MBL, MBO and BBO Data ..... 19

3. Official Closing Price ..... 20

4. Special Behavior ..... 20
   - 4.1. Level1 Market Data Kinematics – Post-Close ..... 20
   - 4.2. Update of the Level1 Market Data Kinematics – CLOSE ..... 22
   - 4.3. Update of the Level1 Market Data Kinematics – Halted Instruments Behavior ..... 23

5. Finding the Latest Information . . . . . 24

# FEEDOS™ OSLO MIT FEED DESCRIPTION

As part of S&P Capital IQ Real-Time Solutions FeedOS documentation, this feed description provides you with details about the types of data broadcast on the OSLO MIT market data stream, their possible values and current FeedOS technical implementation.

The topics this feed description covers include:


- 1. Referential Data
- 2. Quotation Data
- 3. Official Closing Price
- 4. Special Behavior
- 5. Finding the Latest Information.


# 1. Referential Data

The following sections describe the characteristics of the referential data on the OSLO MIT market data stream, in terms of:


- 1.1. Available Markets and Branches
- 1.2. Types of Instruments
- 1.3. Specific Referential Tags.


## 1.1. Available Markets and Branches

This section details the list of markets and branches available on the OSLO MIT market data stream:


- 1.1.1. Markets
- 1.1.2. Branches.

### 1.1.1. Markets

The OSLO MIT market data stream broadcasts informations about the following markets:

**Table 1 List of markets available on OSLO MIT market data stream**

| FeedOS Market ID | Market |
|---|---|
| XOSL | Oslo Børs |
| BURG | Burgundy Nordic MTF |
| XOAM | Oslo Børs Alternative Bond Market |
| XOAS | Oslo Axess |

The following example shows the list of markets available on OSLO MIT market data stream and their IDs, returned by the command `dumps`:

```
MARKETS
market # 187    CC=NO/NORWAY/OLSO,DESCR=OSLO BORS, WEB=www.ose.no
    MIC = XOSL
    TimeZone = Europe/Oslo
    Country = NO
    NbMaxInstruments = 2000000
market # 259    CC=SE/SWEDEN/STOCKHOLM,DESCR=BURGUNDY NORDIC MTF, WEB=www.burgundy.se
    MIC = BURG
    TimeZone = Europe/Oslo
    Country = NO
    NbMaxInstruments = 2000000
market # 437    CC=NO/NORWAY/OLSO,DESCR=OSLO BORS ALTERNATIVE BOND MARKET,
WEB=www.abmportal.no
    MIC = XOAM
    TimeZone = Europe/Oslo
    Country = NO
    NbMaxInstruments = 2000000
market # 439    CC=NO/NORWAY/OLSO,DESCR=OSLO AXESS, WEB=www.osloaxess.no
    MIC = XOAS
    TimeZone = Europe/Oslo
    Country = NO
    NbMaxInstruments = 2000000
```

### 1.1.2. Branches

The example below shows the list of branches available on OSLO MIT market data stream, returned by the command `dumps`. Each branch displays the following details: `FOSMarketID`, `SecurityType`, `CFICode` and `Quantity` (of instruments):

```
BRANCHES
    { XOSL CS   ESXXXX } qty: 201
    { XOSL NONE EUXXXX } qty: 129
    { XOSL NONE EXXXXX } qty: 21
    { XOSL NONE RSXXXX } qty: 20
    { XOSL WAR  RWXXXX } qty: 1183
    { BURG CS   ESXXXX } qty: 996
    { BURG NONE EUXXXX } qty: 80
    { BURG NONE RSXXXX } qty: 14
    { XOAM NONE RXXXXX } qty: 92
    { XOAS CS   ESXXXX } qty: 45
    { XOAS NONE RSXXXX } qty: 5
```

## 1.2. Types of Instruments

The following sections illustrate the instruments' characteristics on OSLO MIT market data stream, according to their type:


- 1.2.1. Equities
- 1.2.2. Rights
- 1.2.3. Warrants.


### 1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 187/756990 = 392924414
    PriceCurrency               string{NOK}
    Symbol                      string{NANO}
    Issuer                      string{Nordic Nanovector ASA}
    Description                 string{Nordic Nanovector}
    SecurityType                string{CS}
    FOSMarketId                 XOSL
    CFICode                     string{ESXXXX}
    CountryOfIssue              string{NO}
    RoundLot                    float64{1}
    SecuritySubType             string{SH}
    SecurityGroup               string{OBNW}
    InternalCreationDate        Timestamp{2015-03-23 02:01:02:231}
    InternalModificationDate    Timestamp{2015-04-02 01:01:02:418}
    InternalSourceId            uint16{65}
    LocalCodeStr                string{1301592}
    ISIN                        string{NO0010597883}
    PriceIncrement_dynamic_TableId      uint32{4259941}
    OperatingMIC                string{XOSL}
    DynamicVariationRange       float64{0}
    StaticVariationRange        float64{0}
    MARKET_LSE_NormalMarketSize float64{300}
    MARKET_LSE_SegmentCode      string{OBNW}
```

### 1.2.2. Rights

The sample below illustrates the details of a right:

```
instr # 187/756626 = 392924050
    PriceCurrency               string{NOK}
    Symbol                      string{NTS S}
    Issuer                      string{NTS ASA}
    Description                 string{NTS TR}
    SecurityType                string{NONE}
    FOSMarketId                 XOSL
    CFICode                     string{RSXXXX}
    CountryOfIssue              string{NO}
    RoundLot                    float64{1}
    SecuritySubType             string{RG}
    SecurityGroup               string{OBST}
    InternalCreationDate        Timestamp{2014-11-20 02:01:02:086}
    InternalModificationDate    Timestamp{2015-01-08 09:11:56:287}
    InternalSourceId            uint16{65}
    InternalAggregationId       uint16{65}
    InternalEntitlementId       int32{1084}
    DelayedFeedMin              uint16{15}
    LocalCodeStr                string{1301389}
    ISIN                        string{NO0010722671}
    MaturityYear                uint16{2014}
    MaturityMonth               uint8{12}
    MaturityDay                 uint8{4}
    PriceIncrement_dynamic_TableId      uint32{4259941}
    OperatingMIC                string{XOSL}
    DynamicVariationRange       float64{900}
    StaticVariationRange        float64{900}
    MARKET_LSE_NormalMarketSize float64{500}
    MARKET_LSE_SegmentCode      string{OBST}
```

### 1.2.3. Warrants

The sample below illustrates the details of a warrant:

```
instr # 187/756967 = 392924391
    PriceCurrency               string{NOK}
    Symbol                      string{YARW15I480HA}
    Issuer                      string{SVENSKA HANDELSBANKEN AB}
    Description                 string{YARW15I480HA}
    SecurityType                string{WAR}
    FOSMarketId                 XOSL
    CFICode                     string{RWXXXX}
    CountryOfIssue              string{NO}
    RoundLot                    float64{1}
    SecuritySubType             string{WA}
    SecurityGroup               string{OBWR}
    InternalCreationDate        Timestamp{2015-02-18 02:01:02:277}
    InternalModificationDate    Timestamp{2015-04-02 04:45:00:142}
    InternalSourceId            uint16{65}
    InternalAggregationId       uint16{65}
    InternalEntitlementId       int32{1084}
    DelayedFeedMin              uint16{15}
    LocalCodeStr                string{2015341}
    ISIN                        string{NO0010731482}
    MaturityYear                uint16{2015}
    MaturityMonth               uint8{9}
    MaturityDay                 uint8{18}
    PriceIncrement_dynamic_TableId      uint32{4259941}
    OperatingMIC                string{XOSL}
    DynamicVariationRange       float64{0}
    StaticVariationRange        float64{0}
    MARKET_LSE_NormalMarketSize float64{10000}
    MARKET_LSE_SegmentCode      string{OBWR}
```

## 1.3. Specific Referential Tags

The following sections describe additional, specific referential tags available on the OSLO MIT market data stream:


- 1.3.1. SecurityStatus
- 1.3.2. OperatingMIC and SegmentMIC
- 1.3.3. DynamicVariationRange
- 1.3.4. StaticVariationRange
- 1.3.5. MARKET_LSE_NormalMarketSize
- 1.3.6. MARKET_LSE_SegmentCode.


### 1.3.1. SecurityStatus

The values of the referential tag **SecurityStatus** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Referential* to indicate the status of an instrument.

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

**Table 2 SecurityStatus – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `SecurityStatus` | FeedOS tag name. |
| **Numeric ID** | `965` | 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** | *`[Exchange Specific Value]`* | An ***exchange specific value***, indicating the status of an instrument. |
| **Possible Values** | `1` | Active (Default value) |
| | `2` | Inactive |
| | `3` | Suspended |

## 1.3.2. OperatingMIC and SegmentMIC

The values of the referential tags **OperatingMIC** and **SegmentMIC** conveyed on the OSLO MIT market data stream are disseminated via FeedOS in *Referential* to reflect the exchange adoption of the ISO 10383:2012 standard. This new edition of the ISO standard refines the level of granularity on OSLO MIT market data stream, by introducing two levels of MIC codes – *operating* (parent-like) and *market segment* (child-like) MICs.

FeedOS implementation of the tags `OperatingMIC` and `SegmentMIC` is described in the following table:

**Table 3 OperatingMIC and SegmentMIC – technical implementation in FeedOS**

| Component | Value | | Description |
|---|---|---|---|
| **Tag Name** | `OperatingMIC` | `SegmentMIC` | FeedOS tag name. |
| **Numeric ID** | `9533` | `9534` | 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` | String data type. |
| **Format** | *`[Exchange Specific Value]`* | *`[Exchange Specific Value]`* | An ***exchange specific value***, specifying the parent and child MICs. |
| **Possible Values** | `XOSL` | `BURG` | BURGUNDY NORDIC MTF |
| | | `XOAM` | NORDIC ALTERNATIVE BOND MARKET |
| | | `XOAS` | OSLO AXESS |

## 1.3.3. DynamicVariationRange

The values of the referential tag **DynamicVariationRange** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Referential* to indicate the maximum permitted value around the dynamic price.

The **Dynamic Range** defines the maximum permitted variation around the *Dynamic Price* (in both directions) and it is expressed as a percentage. The *Dynamic Price* is the price fixed *in the last trade*, and may be the result either of an auction (in which case it will be the same as the static price) or of a trade made on the open market. The Dynamic Range remains in force only while the market is open and during the closing auction.

S&P Capital IQ Real-Time Solutions disseminates only the variation ranges related to the continuous trading session.

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

**Table 4 DynamicVariationRange – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `DynamicVariationRange` | FeedOS tag name. |
| **Numeric ID** | `9553` | 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 percentile value***, detailing the maximum permitted value around the dynamic price. |

## 1.3.4. StaticVariationRange

The values of the referential tag **StaticVariationRange** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Referential* to indicate the maximum permitted value around the static price.

The **Static Range** defines the maximum permitted variation around the *Static Price* (in both directions) and it is expressed as a percentage. The *Static Price* is the price fixed *at the last auction* (the auction allocation price). The Static Range remains in force during the entire session.

S&P Capital IQ Real-Time Solutions disseminates only the variation ranges related to the continuous trading session.

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

**Table 5 StaticVariationRange – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `StaticVariationRange` | FeedOS tag name. |
| **Numeric ID** | `9554` | 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 percentile value***, detailing the maximum permitted value around the static price. |

## 1.3.5. MARKET_LSE_NormalMarketSize

The values of the referential tag **MARKET_LSE_NormalMarketSize** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Referential* to detail the size of the transaction.

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

**Table 6 MARKET_LSE_NormalMarketSize – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_NormalMarketSize` | FeedOS tag name. |
| **Numeric ID** | 11000 | 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 size of the transaction. |

## 1.3.6. MARKET_LSE_SegmentCode

The values of the referential tag **MARKET_LSE_SegmentCode** conveyed on the OSLO MIT market data stream are disseminated via FeedOS market data stream in *Referential* to uniquely identify a specific trading area as defined by Oslo Børs.

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

**Table 7 MARKET_LSE_SegmentCode – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_SegmentCode` | FeedOS tag name. |
| **Numeric ID** | 11002 | 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***, uniquely identifying a specific trading area, as described below. |
| **Possible Values** | `OAX` | The OAX segment includes the Oslo Axess listed securities. This does not imply any change in Oslo Axess' status as a regulated market. |
| | `OBEF` | The OBEF segment includes the listed ETFs (Exchange Traded Funds). OBEN The OBEN segment includes the listed ETNs (Exchange Traded Notes). |
| | `OBMA` | The OBMA segment includes the securities in liquidity category OB Match, such as shares with a minimum of 10 trades per day on average, excl. OBX or shares with less than 10 trades but with a liquidity provider scheme. |
| | `OBNW` | The OBNW segment includes the securities in liquidity category OB New, such as newly listed shares. |
| | `OBPC` | The OBPC segment includes the securities in the liquidity category OB Equity Certificates, such as the listed equity certificates. |

**Table 7 MARKET_LSE_SegmentCode – technical implementation in FeedOS (Continued)**

| Component | Value | Description |
|---|---|---|
| **Possible Values** | OBST | The OBST segment includes the securities in liquidity category OB Standard, such as shares with fewer than 10 trades per day on average and without a liquidity provider agreement. |
| | OBTS | This is a segment including instruments with different instrument types. The details are described in *OSLMIT 501 Guide to Testing Services3*.<br>Please notice that access to these instruments may be restricted by Oslo Børs without any prior notice. |
| | OBWR | The OBWR segment includes warrants. |
| | OBX | This segment includes the constituents of the OBX index – the most traded stocks at Oslo Børs. |

# 2. Quotation Data

The following sections describe the characteristics of the quotation data on OSLO MIT market data stream, in terms of:


- 2.1. Quotation Values
- 2.2. TradingStatus
- 2.3. Specific Quotation Tags
- 2.4. MBL, MBO and BBO Data.

## 2.1. Quotation Values

The examples below shows the possible values of an instrument on OSLO MIT market data stream:

```
InstrumentStatusL1
-- 187/756990
        BID: 34.7       6016    @1
        ASK: 34.8       2862    @2
        LastPrice                       float64{34.7}
        LastTradeQty                    float64{100}
        DailyHighPrice                  float64{35.5}
        DailyLowPrice                   float64{34}
        DailyTotalVolumeTraded          float64{276752}
        DailyTotalAssetTraded           float64{9571516.7}
        LastTradePrice                  float64{34.7}
        LastTradeTimestamp              Timestamp{2015-03-24 10:32:13:409}
        InternalDailyOpenTimestamp      Timestamp{2015-03-24 08:00:29:015}
        InternalDailyCloseTimestamp     Timestamp{2015-03-23 16:37:00:084}
        InternalDailyHighTimestamp      Timestamp{2015-03-24 09:13:46:004}
        InternalDailyLowTimestamp       Timestamp{2015-03-24 09:01:48:613}
        InternalPriceActivityTimestamp  Timestamp{2015-03-24 10:32:13:409}
        TradingStatus                   17=ReadyToTrade
        LastOffBookTradePrice           float64{34.6}
        LastOffBookTradeQty             float64{7835}
        LastOffBookTradeTimestamp       Timestamp{2015-03-24 09:30:54:747}
        DailyOpeningPrice               float64{34.7}
        PreviousDailyTotalVolumeTraded  float64{2173729}
        PreviousDailyTotalAssetTraded   float64{75511846.2}
        PreviousDailyClosingPrice       float64{34.5}
        PreviousBusinessDay             Timestamp{2015-03-23}
        CurrentBusinessDay              Timestamp{2015-03-24}
        LastAuctionPrice                float64{34.7}
        LastAuctionVolume               float64{8150}
        DailyTotalOffBookVolumeTraded   float64{23505}
        DailyTotalOffBookAssetTraded    float64{813273}
        PreviousInternalDailyClosingPriceType   char{a}
        InternalLastAuctionTimestamp    Timestamp{2015-03-24 08:00:26:797}
        PriceActivityMarketTimestamp    Timestamp{2015-03-24 10:32:13:409}
        MARKET_TURQUOISE_DarkBookTradingStatus  Enum{17}
        MARKET_TURQUOISE_OffBookReportingTradingStatus  Enum{17}
        MARKET_LSE_MIT_TradingStatusDetails     char{T}
        MARKET_LSE_MIT_TotalAuctionVolume       float64{8150}
```

For more details about the fields and tags available in quotation data type, and their possible values, see *FeedOS Quotation Tags Guide*.

## 2.2. TradingStatus

Each time a modification of the trading status occurs, the values of the quotation tag **TradingStatus** conveyed on the OSLO MIT 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:

**Table 8 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's 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 |

## 2.3. Specific Quotation Tags

The following sections describe additional, specific quotation tags available on OSLO MIT market data stream:

- 2.3.1. Trade Conditions
- 2.3.2. Other Values.

### 2.3.1. Trade Conditions

The following subsections describe the trade conditions on OSLO MIT market data stream:

- 2.3.1.1. TradeID
- 2.3.1.2. MMTFlagsV2
- 2.3.1.3. Aggressor Side
- 2.3.1.4. MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator
- 2.3.1.5. MARKET_LSE_MIT_AuctionTypeIndicator.

#### 2.3.1.1. TradeID

Each time a trade occurs, the values of the quotation tag **TradeID** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Context* to identify the 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 values currently available for the tag `TradeID` is described in the following table:

**Table 9 TradeID – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `TradeID` | FeedOS tag name. |
| **Numeric ID** | `1003` | 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*** identifying the trade. |

### 2.3.1.2. MMTFlagsV2

Each time a trade occurs, the values of the quotation tag **MMTFlagsV2** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Context* to detail the Market Model Typology (version 2) applicable to the 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 `MMTFlagsV2` is described in the table below (newly available values are in `green`):

**Table 10 MMTFlagsV2 – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MMTFlagsV2` | FeedOS tag name. |
| **Numeric ID** | `9901` | 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]`* *`10-character long`* | An ***exchange specific value***, detailing the Market Model Typology (version 2) applicable to the trade. |
| **Possible Values** | **MMT Level 1 - MARKET MECHANISM – OFFSET 1** | |
| | `1` | Central Limit Order Book |
| | `2` | Quote Driven Market |
| | `3` | Dark Order Book |
| | `4` | Off Book |
| | **MMT Level 2 - TRADING MODE – OFFSET 2** | |
| | `1` | Undefined Auction |
| | `2` | Continuous Trading |
| | `3` | At Market Close Trading |
| | `4` | Out of Main Session Trading |
| | `5` | Trade Reporting (On Exchange) |
| | `6` | Trade Reporting (Off Exchange) |
| | `7` | Trade Reporting (Systematic Internaliser) |
| | `O` | Scheduled Opening Auction |
| | `K` | Scheduled Closing Auction |

**Table 10 MMTFlagsV2 – technical implementation in FeedOS (Continued)**

| Component | Value | Description |
|---|---|---|
| Possible Values | I | Scheduled Intraday Auction |
| | U | Unscheduled Auction |
| | **MMT Level 3 - TRANSACTION TYPE** | |
| | ***3.1. TRANSACTION CATEGORY – OFFSET 3*** | |
| | B | Benchmark Trade |
| | P | Plain-Vanilla Trade |
| | D | Dark Trade |
| | T | Technical Trade |
| | G | Give-up/Give-In Trade |
| | F | Trade with Conditions |
| | ***3.2. NEGOTIATED TRANSACTION INDICATOR – OFFSET 4*** | |
| | N | Negotiated Trade |
| | - | No Negotiated Trade |
| | ***3.3. CROSSING TRADE INDICATOR – OFFSET 5*** | |
| | X | Crossing Trade |
| | - | No Crossing Trade |
| | ***3.4. MODIFICATION INDICATOR – OFFSET 6*** | |
| | C | Trade Cancellation |
| | A | Trade Amendment |
| | - | New Trade |
| | ***3.5. BENCHMARK INDICATOR – OFFSET 7*** | |
| | B | Benchmark Trade |
| | - | No Benchmark Trade |
| | ***3.6. EX/CUM DIVIDEND INDICATOR – OFFSET 8*** | |
| | E | Ex/cum dividend Trade |
| | - | No Ex/Cum Dividend Trade |
| | **MMT Level 4 - PUBLICATION MODE – OFFSET 9** | |
| | - | Immediate Publication |
| | 1 | Non Immediate Publication |

### 2.3.1.3. Aggressor Side

Each time a trade occurs, the values of the quotation context tag **Aggressor Side** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Context*, to indicate whether the aggressor is a buyer or a seller:

- 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 `AggressorSide` is described in the following table:

**Table 11 AggressorSide – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `AggressorSide` | FeedOS tag name. |
| **Numeric ID** | `9356` | 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***, indicating whether the aggressor is a buyer or a seller. |
| **Possible Values** | `Space` | No aggressor |
| | `1` | Buy Side |
| | `2` | Seller Side |

### 2.3.1.4. MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator

Each time a trade occurs, the values of the quotation tag **MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Context* to detail the off book 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 `MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator` is described in the table below:

**Table 12 MARKET_LSE_MIT_OffBookReportingTradeTypeIndicator – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_MIT_OffBookReportingTrade TypeIndicator` | FeedOS tag name. |
| **Numeric ID** | `15950` | 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** | *`[Exchange Specific Value]`* | An ***exchange specific value***, detailing the off book trade type. |
| **Possible Values** | `20` | NM – Not to Mark |
| | `1000` | O – Ordinary trade |
| | `1005` | NK – Negotiated trade – delayed publ. requested |
| | `1006` | NT – Negotiated trade |
| | `1008` | OK – Ordinary trade – delayed publ. requested |
| | `1009` | OT – OTC trade |
| | `1011` | SI – SI Trade |
| | `1012` | SK – SI Trade – delayed publ. requested |
| | `1013` | TK – OTC trade – delayed publ. requested |
| | `3012` | DT – Derivative Related |
| | `3013` | VW – VWAP Trade |

#### 2.3.1.5. MARKET_LSE_MIT_AuctionTypeIndicator

Each time a trade occurs, the values of the quotation tag **MARKET_LSE_MIT_AuctionTypeIndicator** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Context* to detail the auction 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 `MARKET_LSE_MIT_AuctionTypeIndicator` is described in the table below:

**Table 13 MARKET_LSE_MIT_AuctionTypeIndicator – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_MIT_AuctionTypeIndicator` | FeedOS tag name. |
| **Numeric ID** | `15951` | 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** | *`[Exchange Specific Value]`* | An ***exchange specific value***, detailing the auction type. |
| **Possible Values** | `A` | Re-Opening Auction |
| | `C` | Closing Auction |
| | `I` | Intra-Day Auction |
| | `O` | Opening Auction |

## 2.3.2. Other Values

The following subsections describe the trade conditions on the OSLO MIT market data stream:


- 2.3.2.1. InternalDailyClosingPriceType
- 2.3.2.2. MARKET_TURQUOISE_DarkBookTradingStatus
- 2.3.2.3. MARKET_TURQUOISE_OffBookReportingTradingStatus
- 2.3.2.4. MARKET_LSE_MIT_TradingStatusDetails
- 2.3.2.5. MARKET_LSE_MIT_HaltReason
- 2.3.2.6. MARKET_LSE_MIT_TotalAuctionVolume.


#### 2.3.2.1. InternalDailyClosingPriceType

The values of the quotation tag **InternalDailyClosingPriceType** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the type of the internal daily closing 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 `InternalDailyClosingPriceType` is described in the table below (currently disseminated values are in `green`):

**Table 14 InternalDailyClosingPriceType – technical implementation in QuantFEED®**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `InternalDailyClosingPriceType` | FeedOS tag name. |
| **Numeric ID** | `9155` | 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** | *`[Internal Specific Value]`* | An ***internal specific value***, detailing the type of daily closing price, as described below. |
| **Possible Values** | 0 | **Undefined** |
| | `a` | **Official Close** – Explicit closing price value calculated and distributed by an exchange for the main trading session of a given trading day. |
| | `b` | **Official Indicative** – Exchange has provided an indicative price and marked it as indicative, however no trading activity is observed. |
| | `c` | **Official Carry Over** – Explicit Closing price value from a previous trading day carried forward by the exchange to the given trading day. |
| | `d` | **Last Price** – Final price disseminated by the exchange for the main trading session or dissemination period of a given trading day (for indices). |
| | e | **Last Eligible Price** – Execution price of the final trade (subject to trade qualifiers) accepted by the exchange for the main trading session of a given trading day. |
| | z | **Manual** – Price disseminated manually (in case of production correction). |

### 2.3.2.2. MARKET_TURQUOISE_DarkBookTradingStatus

The values of the quotation tag **MARKET_TURQUOISE_DarkBookTradingStatus** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the trading status:

- 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_TURQUOISE_DarkBookTradingStatus` is described in the table below:

**Table 15 MARKET_TURQUOISE_DarkBookTradingStatus – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_TURQUOISE_DarkBookTradingStatus` | FeedOS tag name. |
| **Numeric ID** | `14721` | 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** | `Enum` | Enum data type. |
| **Format** | *`[Exchange Specific Value]`* | An ***exchange specific value***, indicating the trading status of the Dark Book. |

**Table 15 MARKET_TURQUOISE_DarkBookTradingStatus – technical implementation in FeedOS (Continued)**

| Component | Value | Description |
|---|---|---|
| **Possible Values** | 2 | Trading Halt |
| | 5 | Price Indication |
| | 17 | Ready to Trade |
| | 18 | Not Available for Trading |

### 2.3.2.3. MARKET_TURQUOISE_OffBookReportingTradingStatus

The values of the quotation tag **MARKET_TURQUOISE_OffBookReportingTradingStatus** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the off book trading status:

- 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_TURQUOISE_OffBookReportingTradingStatus` is described in the table below:

**Table 16 MARKET_TURQUOISE_OffBookReportingTradingStatus – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_TURQUOISE_OffBookReporting TradingStatus` | FeedOS tag name. |
| **Numeric ID** | `14722` | 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** | `Enum` | Enum data type. |
| **Format** | *`[Exchange Specific Value]`* | An ***exchange specific value***, indicating the off book trading status. |
| **Possible Values** | 2 | Trading Halt |
| | 17 | Ready to Trade |
| | 18 | Not Available for Trading |

### 2.3.2.4. MARKET_LSE_MIT_TradingStatusDetails

Each time a modification of the instrument status occurs, the values of the quotation tag **MARKET_LSE_MIT_TradingStatusDetails** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the current status of the instrument:

- 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_LSE_MIT_TradingStatusDetails` is described in the table below:

**Table 17 MARKET_LSE_MIT_TradingStatusDetails – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_MIT_TradingStatusDetails` | FeedOS tag name. |
| **Numeric ID** | `14750` | 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** | *`[Exchange Specific Value]`* | An ***exchange specific value***, detailing the current status of an instrument. |
| **Possible Values** | `2` | Intra-day Auction |
| | `H` | Halt |
| | `T` | Regular Trading/Start of Trade Reporting |
| | `a` | Opening Auction Call |
| | `b` | Post-Close |
| | `c` | Market Close |
| | `d` | Closing Auction Call |
| | `e` | Re-Opening Auction Call |
| | `v` | End of Trade Reporting |
| | `w` | No Active Session |
| | `x` | End of Post Close |
| | `y` | Pre-Trading (Start of Trading) |
| | `z` | Closing Price Publication |

### 2.3.2.5. MARKET_LSE_MIT_HaltReason

Each time an instrument is halted from trading, the values of the quotation tag **Halt Reason** conveyed on the OSLO MIT 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_LSE_MIT_HaltReason` is described in the table below:

**Table 18 MARKET_LSE_MIT_HaltReason – technical implementation in FeedOS**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_MIT_HaltReason` | FeedOS tag name. |
| **Numeric ID** | `14752` | 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** | *`[Exchange Specific Value]`* | An ***exchange specific value***, detailing the reason of halting an instrument. |

**Table 18 MARKET_LSE_MIT_HaltReason – technical implementation in FeedOS (Continued)**

| Component | Value | Description |
|---|---|---|
| **Possible Values** | 1 | Price movement |
| | 2 | Received announcement |
| | 3 | In anticipation of announcement |
| | 4 | System problems |
| | 5 | Other |
| | 6 | Reference data update |
| | 101 | Instrument-level circuit breaker tripped |
| | 9998 | Matching partition suspended |
| | 9999 | System suspended |
| | Space | Reason not available |

When an instrument is no longer halted, the tag `MARKET_LSE_MIT_HaltReason` is reset by sending a value with the syntax `UNKNOWN`.

#### 2.3.2.6. MARKET_LSE_MIT_TotalAuctionVolume

The values of the quotation tag **MARKET_LSE_MIT_TotalAuctionVolume** conveyed on the OSLO MIT market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the volume of the auction:

- 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_LSE_MIT_TotalAuctionVolume` is described in the following table:

**Table 19 MARKET_LSE_MIT_TotalAuctionVolume – technical implementation in QuantFEED®**

| Component | Value | Description |
|---|---|---|
| **Tag Name** | `MARKET_LSE_MIT_TotalAuctionVolume` | FeedOS tag name. |
| **Numeric ID** | 14756 | 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***, indicating the auction's volume. |

## 2.4. MBL, MBO and BBO Data*

The MBL book has a 10-level depth. The MBO book is full depth.

* The MBL, MBO and BBO 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.

# 3. Official Closing Price

The closing price is the last trade price upon close, as provided by the exchange. If the instrument has an auction phase, the market sends the last auction price, which becomes the closing price. When a stock splits, the closing price is adjusted after the closing. There is no settlement price.

# 4. Special Behavior

The following sections describe the special behavior of the OSLO MIT market data stream:


- 4.1. Level1 Market Data Kinematics – Post-Close
- 4.2. Update of the Level1 Market Data Kinematics – CLOSE
- 4.3. Update of the Level1 Market Data Kinematics – Halted Instruments Behavior.


## 4.1. Level1 Market Data Kinematics – Post-Close

In the Post-Close kinematics **before 2014-04-28**, during the Post-Trading Hours (usually, between 16:25 and 17:30 Exchange Standard Time), an instrument had the Trading Status `18=Not Available for Trading`, as shown 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  15:25:00:226.577  392921627  1.33    50000   *       *       *       *
TradeConditionsDictionaryKey=uint32{136314981},HiddenLiquidityFlag=bool{True}
TE  15:25:00:226.675  392921627  *       *       1.329   214000@2        *       *
TE  15:25:00:226.689  392921627  1.33    37280   *       *       1.33    565221@3
VU  15:25:00:226.714  392921627  MARKET_LSE_MIT_TradingStatusDetails=z   TradingStatus=18
SI  15:25:00:233.187  392921627  CLOSE   1.33
TE  15:25:00:233.187  392921627  1.33    *       *       *       *       *       C
VU  15:25:01:025.749  392921627  MARKET_LSE_MIT_TradingStatusDetails=b   TradingStatus=18
VU  16:30:00:056.877  392921627  MARKET_LSE_MIT_TradingStatusDetails=x   TradingStatus=18
TE  16:37:00:150.370  392921627  *       *       1.329   14000@1 *       *
TE  16:37:00:151.794  392921627  *       *       *       *       1.33    67323@2
VU  16:37:00:153.886  392921627  MARKET_LSE_MIT_TradingStatusDetails=c
```

In the Post-Close kinematics **after 2014-04-28**, during the same Post-Trading Hours, an instrument has the Trading Status `15=New Price Indication`, as shown in the next 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  15:25:00:226.577  392921627  1.33    50000   *       *       *       *
TradeConditionsDictionaryKey=uint32{136314981},HiddenLiquidityFlag=bool{True}
TE  15:25:00:226.675  392921627  *       *       1.329   214000@2        *       *
TE  15:25:00:226.689  392921627  1.33    37280   *       *       1.33    565221@3
VU  15:25:00:226.714  392921627  MARKET_LSE_MIT_TradingStatusDetails=z   TradingStatus=18
SI  15:25:00:233.187  392921627  CLOSE   1.33
TE  15:25:00:233.187  392921627  1.33    *       *       *       *       *       C
VU  15:25:01:025.749  392921627  MARKET_LSE_MIT_TradingStatusDetails=b   TradingStatus=15
VU  16:30:00:056.877  392921627  MARKET_LSE_MIT_TradingStatusDetails=x   TradingStatus=18
TE  16:37:00:150.370  392921627  *       *       1.329   14000@1 *       *
TE  16:37:00:151.794  392921627  *       *       *       *       1.33    67323@2
VU  16:37:00:153.886  392921627  MARKET_LSE_MIT_TradingStatusDetails=c
```

## 4.2. Update of the Level1 Market Data Kinematics – CLOSE

In the Level1 Market Data Kinematics **before 2015-04-20**, the CLOSE signal was sent when the `MARKET_LSE_MIT_TradingStatusDetails` received the value `c=MarketClose`, 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   07:10:00:058.199   392918865   MARKET_LSE_MIT_TradingStatusDetails=y
TE   07:10:00:058.215   392918865   *       *       0.2     1000@1  *       *
VU   07:15:00:108.872   392918865   MARKET_LSE_MIT_TradingStatusDetails=a   TradingStatus=21
SI   07:15:00:132.984   392918865   OPEN    *
TE   07:15:00:132.984   392918865   *       *       *       *       *       *       O
VU   07:15:00:132.984   392918865   MARKET_LSE_MIT_TradingStatusDetails=T
InternalDailyClosingPriceType=?
VU   07:45:01:012.758   392918865   MARKET_LSE_MIT_TradingStatusDetails=T
MARKET_LSE_MIT_HaltReason=?
VU   07:45:25:030.183   392918865   TradingStatus=17
TE   09:03:45:533.090   392918865   0.2     1       0.2     999@1   *       *       HL
TradeID=1181374308417942,AggressorSide='2'=Sell,TradeConditionsDictionaryKey=uint32{2663384
05},MMTFlagsV2=12P------
VU   09:03:45:533.090   392918865   DailyOpeningPrice=0.2
TE   09:03:50:573.225   392918865   0.2     1       0.2     998@1   *       *
TradeID=1181374308417944, AggressorSide='2'=Sell,
TradeConditionsDictionaryKey=uint32{266338405}, MMTFlagsV2=12P------
VU   17:20:00:014.078   392918865   MARKET_LSE_MIT_TradingStatusDetails=d
LastAuctionPrice=?      LastAuctionVolume=?     TradingStatus=5
VU   17:25:29:129.044   392918865   MARKET_LSE_MIT_TradingStatusDetails=z   TradingStatus=18
VU   17:25:30:004.886   392918865   MARKET_LSE_MIT_TradingStatusDetails=b   TradingStatus=15
VU   17:58:00:129.422   392918865   MARKET_LSE_MIT_TradingStatusDetails=x   TradingStatus=18
SI   17:59:00:203.602   392918865   CLOSE   0.2
TE   17:59:00:203.602   392918865   0.2     *       *       *       *       *       C
VU   17:59:00:203.602   392918865   MARKET_LSE_MIT_TradingStatusDetails=c
InternalDailyClosingPriceType=d
VU   17:59:00:203.602   392918865   DailyClosingPrice=0.2
```

In the Level1 Market Data Kinematics **after 2015-04-20**, the CLOSE signal will be sent earlier, when the `MARKET_LSE_MIT_TradingStatusDetails` receives the value `z=ClosingPricePublication`, as shown 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   07:10:00:058.199   392918865   MARKET_LSE_MIT_TradingStatusDetails=y
TE   07:10:00:058.215   392918865   *       *       0.2     1000@1  *       *
VU   07:15:00:108.872   392918865   MARKET_LSE_MIT_TradingStatusDetails=a   TradingStatus=21
SI   07:15:00:132.984   392918865   OPEN    *
TE   07:15:00:132.984   392918865   *       *       *       *       *       *       O
VU   07:15:00:132.984   392918865   MARKET_LSE_MIT_TradingStatusDetails=T
MARKET_TURQUOISE_OffBookReportingTradingStatus=17       InternalDailyClosingPriceType=?
VU   07:45:01:012.758   392918865   MARKET_LSE_MIT_TradingStatusDetails=T
MARKET_TURQUOISE_DarkBookTradingStatus=17       MARKET_LSE_MIT_HaltReason=?
VU   07:45:25:030.183   392918865   TradingStatus=17
TE   09:03:45:533.090   392918865   0.2     1       0.2     999@1   *       *       HL
TradeID=1181374308417942, AggressorSide='2'=Sell,
TradeConditionsDictionaryKey=uint32{266338405}, MMTFlagsV2=12P------
VU   09:03:45:533.090   392918865   DailyOpeningPrice=0.2
TE   09:03:50:573.225   392918865   0.2     1       0.2     998@1   *       *
TradeID=1181374308417944,AggressorSide='2'=Sell,TradeConditionsDictionaryKey=uint32{2663384
05},MMTFlagsV2=12P------
VU   17:20:00:014.078   392918865   MARKET_LSE_MIT_TradingStatusDetails=d
LastAuctionPrice=?      LastAuctionVolume=?     TradingStatus=5
SI   17:25:29:129.044   392918865   CLOSE   0.2
TE   17:25:29:129.044   392918865   0.2     *       *       *       *       *       C
VU   17:25:29:129.044   392918865   MARKET_LSE_MIT_TradingStatusDetails=z
InternalDailyClosingPriceType=d TradingStatus=18
VU   17:25:29:131.648   392918865   DailyClosingPrice=0.2   InternalDailyClosingPriceType=a
VU   17:25:30:004.886   392918865   MARKET_LSE_MIT_TradingStatusDetails=b   TradingStatus=15
VU   17:57:00:046.501   392918865   MARKET_TURQUOISE_OffBookReportingTradingStatus=15
VU   17:58:00:129.422   392918865   MARKET_LSE_MIT_TradingStatusDetails=x   TradingStatus=18
VU   17:59:00:203.602   392918865   MARKET_LSE_MIT_TradingStatusDetails=c
MARKET_TURQUOISE_DarkBookTradingStatus=18
VU   17:59:00:329.970   392918865   MARKET_TURQUOISE_OffBookReportingTradingStatus=18
```

## 4.3. Update of the Level1 Market Data Kinematics – Halted Instruments Behavior

In the Level1 Market Data Kinematics **before 2015-04-20**, halted instruments closed (Trading Status `18=Not Available for Trading`) at the end of the trading day, and then reopened (Trading Status `17=Ready to Trade`) at the beginning of a new trading day, like regularly traded instruments, as shown in the example below:

```
VU    07:22:35:400    392921684    LastAuctionPrice=1.28
TE    07:44:21:122    392921684    *       *       !       0       !       0
VU    07:44:21:122    392921684    TradingStatus=18
TE    07:44:21:122    392921684    *       *       !       0       !       0
VU    07:44:21:125    392921684    LastAuctionPrice=?      LastAuctionVolume=?
```

In the Level1 Market Data Kinematics **after 2015-04-20**, halted instruments will remain halted (Trading Status `2=Trading Halt`) during market closing and opening, until they will be traded again, as shown in the example below:

```
VU    07:22:35:400    392921684    LastAuctionPrice=1.28
TE    07:44:21:122    392921684    *       *       !       0       !       0
VU    07:44:21:122    392921684    TradingStatus=2
TE    07:44:21:122    392921684    *       *       !       0       !       0
VU    07:44:21:125    392921684    LastAuctionPrice=?      LastAuctionVolume=?
```

# 5. Finding the Latest Information

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

- E-mail: support@quanthouse.com
- Web: http://support.quanthouse.com.