S&P CAPITAL IQ
McGRAW HILL FINANCIAL

S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

SGX EQUITIES

Reference n°: 20150724 – 26458 – 26685 – 27246

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: SGX EQUITIES
Reference 20150724 – 26458 – 26685 – 27246
July 24, 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

TABLE OF CONTENTS

FEEDOS™ SGX EQUITIES FEED DESCRIPTION

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

The topics this feed description covers include:

1. Referential Data

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

1.1. Available Markets and Branches

This section details the list of markets and branches available on the SGX EQUITIES market data stream:

1.1.1. Markets

The SGX EQUITIES market data stream disseminates informations about the following markets:

Table 1 List of markets available on the SGX EQUITIES market data stream

FeedOS Market ID	Market
XSES	Singapore Exchange

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

```
MARKETS
market # 226    CC=SG/SINGAPORE/SINGAPORE,DESCR=SINGAPORE EXCHANGE,WEB=www.sgx.com
    MIC = XSES
    TimeZone = Asia/Singapore
    Country = SG
    NbMaxInstruments = 2000000
```

> **Note** For details about the specific markets available on SGX Equities, see also the section 1.3.1. Symbol on page 5.

1.1.2. Branches

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

```
BRANCHES
    { XSES CS   ESXXXA } qty: 159
    { XSES CS   ESXXXX } qty: 2790
    { XSES CS   RXXXXX } qty: 396
    { XSES FORWARD MMFXXX } qty: 1725
    { XSES GO   DBXXXX } qty: 2898
    { XSES INDEX   TIXMBX } qty: 67
    { XSES MF   EUXXMX } qty: 21
    { XSES MF   EUXXXE } qty: 288
    { XSES MF   EUXXXX } qty: 177
    { XSES PS   ERXXXX } qty: 27
    { XSES WAR  RWMXCX } qty: 948
    { XSES WAR  RWMXPX } qty: 591
    { XSES WAR  RWXXCX } qty: 1035
    { XSES WAR  RWXXPX } qty: 627
    { XSES WAR  RWXXXX } qty: 249
```

1.2. Types of Instruments

The following sections describe the instruments available on the SGX EQUITIES market data stream, according to their type:

1.2.1. Bonds

The sample below illustrates the details of a bond:

```
instr # 226/1014443 = 474970795
    PriceCurrency               string{USD}
    Symbol                      string{32XB_BD}
    Description                 string{SumitoMBk f180723S}
    SecurityType                string{GO}
    FOSMarketId                 XSES
    ContractMultiplier          float64{1}
    CFICode                     string{DBXXXX}
    RoundLot                    float64{1}
    InternalCreationDate        Timestamp{2015-07-22 00:00:50:417}
    InternalModificationDate    Timestamp{2015-07-22 00:00:50:417}
    InternalSourceId            uint16{221}
    InternalAggregationId       uint16{221}
    InternalEntitlementId       int32{1091}
    LocalCodeStr                string{32XB_BD}
    ISIN                        string{US865622BU72}
    PriceIncrement_static       float64{0.001}
    MBLLayersDesc               string{0,3}
    OperatingMIC                string{XSES}
```

1.2.2. Equities

The sample below illustrates the details of an equity:

```
instr # 226/1014083 = 474970435
    PriceCurrency               string{USD}
    Symbol                      string{AXD_US}
    Description                 string{GDR Advanta US$S}
    SecurityType                string{CS}
    FOSMarketId                 XSES
    ContractMultiplier          float64{1}
    CFICode                     string{ESXXXA}
    RoundLot                    float64{1}
    InternalCreationDate        Timestamp{2015-05-11 00:00:49:920}
    InternalModificationDate    Timestamp{2015-06-29 00:00:50:683}
    InternalSourceId            uint16{221}
    InternalAggregationId       uint16{221}
    InternalEntitlementId       int32{1091}
    LocalCodeStr                string{AXD_US}
    PriceIncrement_dynamic_TableId      uint32{14483556}
    MBLLayersDesc               string{0,3}
    OperatingMIC                string{XSES}
```

1.2.3. Indices

The sample below shows the details of an index:

```
instr # 226/1014351 = 474970703
    PriceCurrency               string{JPY}
    Symbol                      string{_JP14}
    Description                 string{SGX JP ACC INV 1X IDX NR}
    SecurityType                string{INDEX}
    FOSMarketId                 XSES
    ContractMultiplier          float64{1}
    CFICode                     string{TIXMBX}
    RoundLot                    float64{1}
    InternalCreationDate        Timestamp{2015-07-01 00:00:49:326}
    InternalModificationDate    Timestamp{2015-07-01 00:00:49:326}
    InternalSourceId            uint16{221}
    InternalAggregationId       uint16{221}
    InternalEntitlementId       int32{1091}
    LocalCodeStr                string{_JP14}
    PriceIncrement_static       float64{0.0001}
    MBLLayersDesc               string{0,3}
    OperatingMIC                string{XSES}
```

1.2.4. Warrants

The sample below illustrates the details of a warrant:

```
instr # 226/1014305 = 474970657
    PriceCurrency               string{SGD}
    Symbol                      string{BABW_US}
    Description                 string{ChinaGaoxian W180918}
    SecurityType                string{WAR}
    FOSMarketId                 XSES
    ContractMultiplier          float64{1}
    CFICode                     string{RWXXXX}
    RoundLot                    float64{1}
    InternalCreationDate        Timestamp{2015-06-16 00:00:50:603}
    InternalModificationDate    Timestamp{2015-06-30 00:00:50:559}
    InternalSourceId            uint16{221}
    InternalAggregationId       uint16{221}
    InternalEntitlementId       int32{1091}
    LocalCodeStr                string{BABW_US}
    ISIN                        string{SG42Q1000003}
    PriceIncrement_dynamic_TableId      uint32{14483556}
    MBLLayersDesc               string{0,3}
    OperatingMIC                string{XSES}
```

1.2.5. Forwards

The sample below illustrates the details of a forward:

```
instr # 226/1013719 = 474970071
    PriceCurrency               string{SGD}
    Symbol                      string{ARO_US}
    Description                 string{STI ETF.ES.1503}
    SecurityType                string{FORWARD}
    FOSMarketId                 XSES
    CFICode                     string{MMFXXX}
    RoundLot                    float64{1}
    InternalCreationDate        Timestamp{2015-02-19 00:00:50:041}
    InternalModificationDate    Timestamp{2015-06-27 11:14:46:247}
    InternalSourceId            uint16{221}
    InternalAggregationId       uint16{221}
    InternalEntitlementId       int32{1091}
    LocalCodeStr                string{ARO_US}
    ISIN                        string{SG41M3000002}
    PriceIncrement_static       float64{0.01}
    MBLLayersDesc               string{0,3}
    OperatingMIC                string{XSES}
```

1.3. Specific Referential Tags

The following sections describe specific referential tags available on the SGX EQUITIES market data stream:

1.3.1. Symbol

The values of the referential tag **Symbol** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Referential* to identify the specific market.

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

Table 2 Symbol – technical implementation in FeedOS

Component	Value	Description
Tag Name	`Symbol`	FeedOS tag name.
Numeric ID	`55`	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.

Table 2 Symbol – technical implementation in FeedOS (Continued)

Component	Value	Description
Format	*[Exchange Specific Value]*	An ***exchange specific value***, identifying the specific market.
Possible Values	RY	**Ready market**
	BI	**Buying-in market** – the platform where the Central Depository (CDP) conducts buying-in for failed trades against a short clearing member.
	US	**Unit Share market** – this market allows trading for odd lots in any quantity less than the board size.

1.3.2. RoundLot

The values of the referential tag **RoundLot** conveyed on the SGX EQUITIES 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 below:

Table 3 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	*[Exchange Specific Value]*	An ***exchange specific value***, detailing the smallest order that can be placed.
Possible Values	1	Smallest order is 1.
	10	Smallest order is 10.
	100	Smallest order is 100.
	1000	Smallest order is 1000.

1.3.3. OperatingMIC

The values of the referential tag **OperatingMIC** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Referential* to specify the parent MIC.

FeedOS implementation of the tag `OperatingMIC` is described in below:

Table 4 OperatingMIC – technical implementation in FeedOS

Component	Value	Description
Tag Name	OperatingMIC	FeedOS tag name.
Numeric ID	9533	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***, specifying the parent MIC.
Possible Values	XSES	Parent MIC for all SGX EQUITIES branches.

1.3.4. SegmentMIC

The values of the referential tag **SegmentMIC** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Referential* to specify the child MIC.

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

Table 5 SegmentMIC – technical implementation in FeedOS

Component	Value	Description
Tag Name	`SegmentMIC`	FeedOS tag name.
Numeric ID	`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 data type.
Format	***`[Exchange Specific Value]`***	An ***exchange specific value***, specifying the child MIC.
Possible Values	`XSCA`	Singapore Catalist Market

2. Quotation Data

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

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

2.1. Quotation Values

The example below shows the possible values of an instrument on the SGX EQUITIES market data stream:

```
InstrumentStatusL1
-- 226/1011991
        BID: 0.69       900     @1
        ASK: 0.695      75100   @3
        LastPrice                       float64{0.69}
        LastTradeQty                    float64{100}
        DailyHighPrice                  float64{0.695}
        DailyLowPrice                   float64{0.69}
        DailyTotalVolumeTraded          float64{35400}
        DailyTotalAssetTraded           float64{24476}
        LastTradePrice                  float64{0.69}
        LastTradeTimestamp              Timestamp{2015-07-24 08:22:40:552}
        InternalDailyOpenTimestamp      Timestamp{2015-07-24 01:00:00:021}
        InternalDailyCloseTimestamp     Timestamp{2015-07-24 09:06:00:004}
        InternalDailyHighTimestamp      Timestamp{2015-07-24 03:54:43:514}
        InternalDailyLowTimestamp       Timestamp{2015-07-24 01:38:50:435}
        InternalPriceActivityTimestamp  Timestamp{2015-07-24 09:06:00:004}
        TradingStatus                   18=NotAvailableForTrading
        DailyOpeningPrice               float64{0.69}
        DailyClosingPrice               float64{0.69}
        PreviousDailyTotalVolumeTraded  float64{156600}
        PreviousDailyTotalAssetTraded   float64{107546}
        PreviousDailyClosingPrice       float64{0.69}
        PreviousBusinessDay             Timestamp{2015-07-23}
        CurrentBusinessDay              Timestamp{2015-07-24}
        LastAuctionPrice                float64{0.695}
        LastAuctionVolume               float64{0}
        LastAuctionImbalanceSide        char{O}
        LastAuctionImbalanceVolume      float64{0}
        InternalDailyClosingPriceType   char{d}
        InternalLastAuctionTimestamp    Timestamp{2015-07-24 00:57:40:573}
        PriceActivityMarketTimestamp    Timestamp{2015-07-24 08:22:40:552}
        MARKET_OMNET_OMX_TradingStateName       string{CLOSE}
```

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 SGX EQUITIES 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 6 TradingStatus – technical implementation in QuantFEED®

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
	16	Trade Dissemination Time
	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 the SGX EQUITIES 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 the SGX EQUITIES market data stream:

- 2.3.1.1. TradeCondition
- 2.3.1.2. TradeID.

2.3.1.1. TradeCondition

Each time a trade occurs, the values of the quotation tag **Trade Condition** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Context*:

- 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 `TradeCondition` is described in the table below:

Table 7 TradeCondition – technical implementation in FeedOS

Component	Value	Description
Tag Name	`TradeCondition`	FeedOS tag name.
Numeric ID	`277`	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 conditions of a trade.
Possible Values	`0`	**No Condition** – The default condition. It is set when the users are from different participants.
	`2`	**Internal Trade/Crossing** – In case of an internal trade, this trade condition is added in the broadcast. It is set when both users belong to the same participant.

2.3.1.2. TradeID

Each time a trade occurs, the values of the quotation context tag **TradeID** conveyed on the SGX EQUITIES 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 tag `TradeID` is described in the table below:

Table 8 TradeID – technical implementation in FeedOS

Component	Value	Description
Tag Name	`TradeID`	FeedOS tag name.
Numeric ID	`1003`	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 trade.

2.3.2. Other Values

The following subsections describe additional specific quotation tags on the SGX EQUITIES market data stream:

2.3.2.1. DailyClosingPrice

The values of the quotation tag **DailyClosingPrice** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Other Values* to specify the final price at which a security is traded on a given trading day:

- 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 `DailyClosingPrice` is described in the table below:

Table 9 DailyClosingPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`DailyClosingPrice`	FeedOS tag name.
Numeric ID	`9132`	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***, specifying the final price at which a security is traded on a given trading day.

2.3.2.2. LastAuctionImbalanceSide

The values of the quotation tag **LastAuctionImbalanceSide** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the imbalance side of a closing 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 `LastAuctionImbalanceSide` is described in the table below:

Table 10 LastAuctionImbalanceSide – technical implementation in FeedOS

Component	Value	Description
Tag Name	`LastAuctionImbalanceSide`	FeedOS tag name.
Numeric ID	`9151`	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 imbalance side of a closing auction.
Possible Values	`B`	Buy-side imbalance
	`S`	Sell-side imbalance
	`N`	No imbalance (buy side equals sell side)
	`O`	Insufficient orders to calculate.

2.3.2.3. LastAuctionImbalanceVolume

The values of the quotation tag **LastAuctionImbalanceVolume** conveyed on the SGX EQUITIES market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the imbalance volume of a closing 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 `LastAuctionImbalanceVolume` is described in the table below:

Table 11 LastAuctionImbalanceVolume – technical implementation in FeedOS

Component	Value	Description
Tag Name	`LastAuctionImbalanceVolume`	FeedOS tag name.
Numeric ID	`9152`	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 imbalance volume of a closing auction.

2.3.2.4. InternalDailyClosingPriceType

The values of the quotation tag **InternalDailyClosingPriceType** conveyed on the SGX EQUITIES 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 values currently available for the tag `InternalDailyClosingPriceType` is described in the table below (the values currently disseminated are highlighted in `green`):

Table 12 InternalDailyClosingPriceType – technical implementation in FeedOS

Component	Value	Description
Tag Name	`InternalDailyClosingPriceType`	FeedOS tag name.
Numeric ID	`9155`	FeedOS unique ID disseminated on 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.

Table 12 InternalDailyClosingPriceType – technical implementation in FeedOS (Continued)

Component	Value	Description
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).

2.3.2.5. MARKET_OMNET_OMX_TradingStateName

Each time a modification of the trading state occurs, the values of the quotation tag **MARKET_OMNET_OMX_TradingStateName** conveyed on the SGX EQUITIES 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_OMNET_OMX_TradingStateName` is described in the table below:

Table 13 MARKET_OMNET_OMX_TradingStateName – technical implementation in FeedOS

Component	Value	Description	
Tag Name	`MARKET_OMNET_OMX_TradingStateName`	FeedOS tag name.	
Numeric ID	`14800`	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 current state of the trade.	
Possible Values	`PRE-OPEN`	FIXSecurityTradingStatus_PreOpen	Pre-Open
	`PRE-OPEN1`	FIXSecurityTradingStatus_PreOpen	Pre-Open
	`PRE-OPEN2`	FIXSecurityTradingStatus_PreOpen	Pre-Open
	`NON-CANCEL`	FIXSecurityTradingStatus_TradeDisseminationTime	Non-cancellation
	`OPEN`	FIXSecurityTradingStatus_ReadyToTrade	Open

Table 13 MARKET_OMNET_OMX_TradingStateName – technical implementation in FeedOS (Continued)

Component	Value	Description	
Possible Values	PRE-CLOSE	FIXSecurityTradingStatus_PriceIndication	Pre-Closing
	CLOSE	FIXSecurityTradingStatus_NotAvailableForTrading	Close
	CLOSE_MKT	FIXSecurityTradingStatus_NotAvailableForTrading	Closing Market
	CLOSE_	FIXSecurityTradingStatus_NotAvailableForTrading	Close Signal Collect
	DL	FIXSecurityTradingStatus_NotAvailableForTrading	Delisted
	SUSP	FIXSecurityTradingStatus_TradingHalt	Suspended
	ADJUST	FIXSecurityTradingStatus_PriceIndication	Adjusting
	BI	FIXSecurityTradingStatus_NotAvailableForTrading	Trading restricted to Buying-In market only
	BI_OPEN	FIXSecurityTradingStatus_ReadyToTrade	Buying-In Opening
	CIRB	FIXSecurityTradingStatus_ReadyToTrade	Triggered by the trading engine whenever an order would trade outside of the upper and lower circuit breaker limits for stocks under Circuit Breaker monitoring.
	CIRH	FIXSecurityTradingStatus_PriceIndication	Halt due to Circuit Breaker Triggered
	H	FIXSecurityTradingStatus_PriceIndication	Trading Halt
	J	FIXSecurityTradingStatus_PriceIndication	Instrument Session State Adjust
	MCE	FIXSecurityTradingStatus_PriceIndication	Mandatory Call Event has occurred for Callable Bull/Bear Contracts (CBBC).
	PL	FIXSecurityTradingStatus_NotAvailableForTrading	Pending Listing
	_PRV_CLOSE	FIXSecurityTradingStatus_ReadyToTrade	Previous Close
	_PRV_OPEN	FIXSecurityTradingStatus_ReadyToTrade	Previous Open

2.4. MBL and MBO Data*

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

* 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.

3. Closing Price

At the end of the market day, SGX publishes the daily closing price only for the ETFs listed in the Ready market, after the trading phase ends. There is no closing price computed and published in unit share and buying-in markets or for any other securities listed on SGX. Moreover, there is no settlement price.

However, SGX derives the ETF Reference Closing Price for all ETF counters at the end of each business day upon market close at 17:06 (normal day) or 12:36 (for half-day trading).

4. Special Behavior

The following section describe the special behavior of the EURONEXT UTP market data stream:

4.1. Microsecond Timestamp Precision on the Level1 Market Data

In the Level1 Market Data disseminated, the server timestamps displays microsecond units, 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   2015-05-26 08:41:21:961.433   474957956    3.72   5000   *   *   *   *
TradeCondition=2,TradeID=65630128234823799:0,AggressorSide='2'=Sell
TE   2015-05-26 08:43:06:359.468   474957956    3.72   5000   *   *   *   *
TradeCondition=2,TradeID=65630128234823800:0,AggressorSide='1'=Buy
TE   2015-05-26 08:43:47:103.312   474957956    3.72   5000   *   *   *   *
TradeCondition=2,TradeID=65630128234823801:0,AggressorSide='2'=Sell
```

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