S&P CAPITAL IQ
McGRAW HILL FINANCIAL

S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

TSE EQUITIES

Reference n°: 20150729 – 19921 – 26831 – 27232

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: TSE EQUITIES
Reference 20150729 – 19921 – 26831 – 27232
July 30, 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™ TSE 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 TSE 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 TSE market data stream, in terms of:

1.1. Available Markets and Branches

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

1.1.1. Markets

The TSE market data stream broadcasts informations about the following markets:

Table 1 List of markets available on the TSE market data stream

FeedOS Market ID	Market
XFKA	Fukuoka Stock Exchange
XNGO	Nagoya Stock Exchange
XSAP	Sapporo Securities Exchange
XTKS	Tokyo Stock Exchange

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

```
MARKETS
market # 135    CC=JP/JAPAN/FUKUOKA,DESCR=FUKUOKA STOCK EXCHANGE, WEB=www.fse.or.jp
    MIC = XFKA
    TimeZone = Asia/Tokyo
    Country = JP
    NbMaxInstruments = 2000000
market # 136    CC=JP/JAPAN/NAGOYA,DESCR=NAGOYA STOCK EXCHANGE, WEB=www.nse.or.jp/e/
index.html
    MIC = XNGO
    TimeZone = Asia/Tokyo
    Country = JP
    NbMaxInstruments = 2000000
market # 142    CC=JP/JAPAN/SAPPORO,DESCR=SAPPORO SECURITIES EXCHANGE,
WEB=www.tokeidai.co.jp/sse
    MIC = XSAP
    TimeZone = Asia/Tokyo
    Country = JP
    NbMaxInstruments = 2000000
market # 147    CC=JP/JAPAN/TOKYO,DESCR=TOKYO STOCK EXCHANGE, WEB=www.tse.or.jp
    MIC = XTKS
    TimeZone = Asia/Tokyo
    Country = JP
    NbMaxInstruments = 2000000
```

Caution	The data disseminated via Nagoya Stock Exchange (XNGO) is not available by default. To include XNGO data and enrich the content of the TSE market data stream, please contact your FeedOS project manager.

1.1.2. Branches

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

```
BRANCHES
    { XFKA CS   ESXXXX } qty: 118
    { XFKA NONE EXXXXX } qty: 2
    { XSAP CS   ESXXXX } qty: 61
    { XSAP NONE EXXXXX } qty: 1
    { XTKS CB   DCXXXX } qty: 92
    { XTKS CS   ESXXXA } qty: 48
    { XTKS CS   ESXXXX } qty: 14164
    { XTKS INDEX TIXXXX } qty: 1118
    { XTKS NONE EXXXXX } qty: 1076
    { XTKS PS   EPXXXX } qty: 12
    { XTKS WAR  RWXXXX } qty: 44
```

1.2. Types of Instruments

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

- 1.2.1. Equities
- 1.2.2. Bonds
- 1.2.3. Indices
- 1.2.4. Warrants.

1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 147/1040849 = 309322193
    PriceCurrency               string{JPY}
    Symbol                      string{4980}
    Description                 string{ デクセリアルズ }
    SecurityType                string{CS}
    FOSMarketId                 XTKS
    CFICode                     string{ESXXXX}
    RoundLot                    float64{100}
    SecuritySubType             string{01}
    SecurityGroup               string{0111}
    InternalCreationDate        Timestamp{2015-07-28 09:15:49:333}
    InternalModificationDate    Timestamp{2015-07-28 09:15:49:333}
    InternalSourceId            uint16{235}
    InternalAggregationId       uint16{235}
    InternalEntitlementId       int32{1095}
    LocalCodeStr                string{4980.T3}
    ISIN                        string{JP3548770001}
    OperatingMIC                string{XJPX}
    SegmentMIC                  string{XTK3}
    IndustryCode                string{3200}
```

1.2.2. Bonds

The sample below illustrates the details of a bond:

```
instr # 147/1040845 = 309322189
    PriceCurrency               string{JPY}
    Symbol                      string{900066758}
    Description                 string{ソニー６ ＣＢ}
    SecurityType                string{CB}
    FOSMarketId                 XTKS
    CFICode                     string{DCXXXX}
    RoundLot                    float64{1000000}
    SecuritySubType             string{51}
    SecurityGroup               string{0211}
    InternalCreationDate        Timestamp{2015-07-21 09:15:18:925}
    InternalModificationDate    Timestamp{2015-07-21 09:15:18:925}
    InternalSourceId            uint16{235}
    InternalAggregationId       uint16{235}
    InternalEntitlementId       int32{1095}
    LocalCodeStr                string{900066758.T3}
    ISIN                        string{JP343500PF78}
    OperatingMIC                string{XJPX}
    SegmentMIC                  string{XTK3}
    IndustryCode                string{3650}
```

1.2.3. Indices

The sample below illustrates the details of a index:

```
instr # 147/1040809 = 309322153
    Description                 string{INAV C999}
    SecurityType                string{INDEX}
    FOSMarketId                 XTKS
    CFICode                     string{TIXXXX}
    SecuritySubType             string{INDEX}
    InternalCreationDate        Timestamp{2015-06-27 10:04:54:142}
    InternalModificationDate    Timestamp{2015-06-27 10:04:54:142}
    InternalSourceId            uint16{235}
    InternalAggregationId       uint16{235}
    InternalEntitlementId       int32{1171}
    LocalCodeStr                string{IDX_C999}
    OperatingMIC                string{XJPX}
```

1.2.4. Warrants

The sample below illustrates the details of a warrant:

```
instr # 147/1039637 = 309320981
    PriceCurrency               string{JPY}
    Symbol                      string{38109}
    Description                 string{Ｍ－サイバー Ｓ ２３予}
    SecurityType                string{WAR}
    FOSMarketId                 XTKS
    CFICode                     string{RWXXXX}
    RoundLot                    float64{100}
    SecuritySubType             string{07}
    SecurityGroup               string{0114}
    InternalCreationDate        Timestamp{2015-02-13 09:15:13:499}
    InternalModificationDate    Timestamp{2015-04-08 22:45:00:132}
    InternalHideFromLookup      bool{True}
    InternalSourceId            uint16{235}
    InternalAggregationId       uint16{235}
    LocalCodeStr                string{38109.T3}
    ISIN                        string{JP3311520096}
    OperatingMIC                string{XJPX}
    SegmentMIC                  string{XTK3}
    IndustryCode                string{5250}
```

2. Quotation Data

The following sections describe the characteristics of the quotation data on the TSE 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 examples below shows the possible values of an instrument on the TSE market data stream:

```
InstrumentStatusL1
-- 147/1001000
        BID: 0  0       *NO ORDER*
        ASK: 0  0       *NO ORDER*
        LastPrice                       float64{1633.94}
        DailyHighPrice                  float64{1638.11}
        DailyLowPrice                   float64{1626.27}
        InternalDailyOpenTimestamp      Timestamp{2015-07-29 00:00:00:033}
        InternalDailyCloseTimestamp     Timestamp{2015-07-29 06:00:03:019}
        InternalDailyHighTimestamp      Timestamp{2015-07-29 00:00:16:090}
        InternalDailyLowTimestamp       Timestamp{2015-07-29 01:12:11:077}
        InternalPriceActivityTimestamp  Timestamp{2015-07-29 06:00:03:088}
        TradingStatus                   18=NotAvailableForTrading
        TradingSessionId                int8{2}
        SessionTotalOffBookAssetTraded  float64{0}
        SessionTotalOffBookVolumeTraded float64{0}
        PriorSessionsTotalAssetTraded   float64{0}
        PriorSessionsTotalVolumeTraded  float64{0}
        PriorSessionsTotalOffBookAssetTraded    float64{0}
        PriorSessionsTotalOffBookVolumeTraded   float64{0}
        SessionTotalVolumeTraded        float64{0}
        SessionOpeningPrice             float64{1631.16}
        PreviousSessionClosingPrice     float64{1627.24}
        SessionHighPrice                float64{1636.08}
        SessionLowPrice                 float64{1630.26}
        SessionTotalAssetTraded         float64{0}
        SessionClosingPrice             float64{1634.08}
        DailyOpeningPrice               float64{1637.97}
        DailyClosingPrice               float64{1633.94}
        PreviousDailyClosingPrice       float64{1629.46}
        PreviousBusinessDay             Timestamp{2015-07-28}
        CurrentBusinessDay              Timestamp{2015-07-29}
        InternalDailyClosingPriceType   char{d}
        PriceActivityMarketTimestamp    Timestamp{2015-07-29 06:00:03}
        ExchangeLastComputedPrice       float64{1633.94}
        MARKET_TSE_BidMarketOrderVolume Float64{20180}
        MARKET_TSE_AskMarketOrderVolume Float64{21300}
        MARKET_TSE_BidSpecialQuotePrice Float64{1670}
        MARKET_TSE_AskSpecialQuotePrice Float64{1684}
```

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 TSE 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 2 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
	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 TSE 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 available on the TSE market data stream:

- 2.3.1.1. MARKET_TSE_BidQuoteCondition
- 2.3.1.2. MARKET_TSE_AskQuoteCondition
- 2.3.1.3. MARKET_TSE_TostnetPriceCode
- 2.3.1.4. MARKET_TSE_TostnetTransactionFlag.

2.3.1.1. MARKET_TSE_BidQuoteCondition

Each time an execution occurs, the values of the quotation context tag **MARKET_TSE_BidQuoteCondition** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Context* to detail the quote condition on the bid side:

- 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_TSE_BidQuoteCondition` is described in the table below:

Table 3 MARKET_TSE_BidQuoteCondition – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_BidQuoteCondition`	FeedOS tag name.
Numeric ID	`16390`	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	`Char`	Char data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the quote condition on the bid side.
Possible Values	0	Quote before Opening
	1	General Quote
	3	Special Quote
	4	Continuous Execution Quote
	7	Special Quote before Trading Halt
	8	Continuous Execution Quote before Trading Halt

2.3.1.2. MARKET_TSE_AskQuoteCondition

Each time an execution occurs, the values of the quotation context tag **MARKET_TSE_AskQuoteCondition** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Context* to detail the quote condition on the ask side:

- 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_TSE_AskQuoteCondition` is described in the table below:

Table 4 MARKET_TSE_AskQuoteCondition – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_AskQuoteCondition`	FeedOS tag name.
Numeric ID	`16391`	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	`Char`	Char data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the quote condition on the ask side.
Possible Values	0	Quote before Opening
	1	General Quote
	3	Special Quote
	4	Continuous Execution Quote
	7	Special Quote before Trading Halt
	8	Continuous Execution Quote before Trading Halt

2.3.1.3. MARKET_TSE_TostnetPriceCode

Each time an execution occurs, the values of the quotation context tag **MARKET_TSE_TostnetPriceCode** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Context* to detail the code of the Tostnet 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 tag `MARKET_TSE_TostnetPriceCode` is described in the table below:

Table 5 MARKET_TSE_TostnetPriceCode – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_TostnetPriceCode`	FeedOS tag name.
Numeric ID	`16392`	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	`UInt8`	UInt8 data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the code of the Tostnet Price.
Possible Values	`<empty>`	Default value, not sent.
	`01`	Previous-day closing price
	`05`	Previous-day VWAP
	`11`	Morning-session closing price
	`15`	Morning-session VWAP
	`31`	Today's closing price
	`25`	Afternoon-session VWAP
	`35`	All-day VWAP

2.3.1.4. MARKET_TSE_TostnetTransactionFlag

Each time an execution occurs, the values of the quotation context tag **MARKET_TSE_TostnetTransactionFlag** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Context* to identify the flag of the Tostnet transaction:

- 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_TSE_TostnetTransactionFlag` is described in the table below:

Table 6 MARKET_TSE_TostnetTransactionFlag – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_TostnetTransactionFlag`	FeedOS tag name.
Numeric ID	`16393`	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	`Char`	Char data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, identifying the flag of the Tostnet transaction.

Table 6 MARKET_TSE_TostnetTransactionFlag – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	<empty>	Ordinary transaction – default value, not sent.
	1	Day transaction
	2	Ordinary transaction (VWAP transaction)
	3	Day transaction (VWAP transaction)

2.3.2. Other Values

The following subsections describe the other values available on the TSE market data stream:

- 2.3.2.1. RegSHOAction
- 2.3.2.2. InternalDailyClosingPriceType
- 2.3.2.3. ExchangeLastComputedPrice
- 2.3.2.4. MARKET_TSE_BidMarketOrderVolume
- 2.3.2.5. MARKET_TSE_AskMarketOrderVolume
- 2.3.2.6. MARKET_TSE_BidSpecialQuotePrice
- 2.3.2.7. MARKET_TSE_AskSpecialQuotePrice.

2.3.2.1. RegSHOAction

Each time a short sale price restriction occurs, the values of the quotation tag **RegSHOAction** conveyed on the TSE 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 `RegSHOAction` is described in the following table:

Table 7 RegSHOAction – technical implementation in FeedOS

Component	Value	Description
Tag Name	`RegSHOAction`	FeedOS tag name.
Numeric ID	9113	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 short sale restriction status.
Possible Values	`FOSRegSHOAction_NoPriceTest`	Short Selling Regulation Does Not Apply (Exchange's Short Selling Regulation = 0)
	`FOSRegSHOAction_PriceTestInEffect`	Short Selling Regulation Is Effective (Exchange's Short Selling Regulation = 1)

2.3.2.2. InternalDailyClosingPriceType

The values of the quotation tag **InternalDailyClosingPriceType** conveyed on the TSE 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 (the values currently disseminated are highlighted in `green`):

Table 8 InternalDailyClosingPriceType – technical implementation in FeedOS

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).

2.3.2.3. ExchangeLastComputedPrice

The values of the quotation tag **ExchangeLastComputedPrice** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the last computed price sent by the exchange:

- 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.

The `ExchangeLastComputedPrice` is available only for certain indices that are eligible for the High Speed Index (during the Continuous Trading phases of the TOPIX / TOPIX Core30 and TOPIX 500).

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

Table 9 ExchangeLastComputedPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`ExchangeLastComputedPrice`	FeedOS tag name.
Numeric ID	`9371`	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 last computed price sent by the exchange.

2.3.2.4. MARKET_TSE_BidMarketOrderVolume

The values of the quotation tag **MARKET_TSE_BidMarketOrderVolume** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Other Values* to detail the market order volume on the bid side:

- 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.

During the auction phases, the TSE Standard Flex Feed does not provide any Theoretical Opening or Auction Price, especially when Market orders are available. Moreover, the Market orders are the first to be executed.

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

Table 10 MARKET_TSE_BidMarketOrderVolume – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_BidMarketOrderVolume`	FeedOS tag name.
Numeric ID	`15060`	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 market order volume on the bid side.

2.3.2.5. MARKET_TSE_AskMarketOrderVolume

The values of the quotation tag **MARKET_TSE_AskMarketOrderVolume** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Other Values* to detail the market order volume on the ask side:

- 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.

During the auction phases, the TSE Standard Flex Feed does not provide any Theoretical Opening or Auction Price, especially when Market orders are available. Moreover, the Market orders are the first to be executed.

Following the introduction of the `MARKET_TSE_BidMarketOrderVolume` and `MARKET_TSE_AskMarketOrderVolume` tags, the Market Order Prices (Magical Prices) are no longer provided in the L1. Moreover, only these tags disseminate the information describing the volume of market orders. However, the MBL L2 does not change, as the markets orders remain in the first position. Subsequently, the L1 and the first limit of the MBL L2 may differ.

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

Table 11 MARKET_TSE_AskMarketOrderVolume – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_AskMarketOrderVolume`	FeedOS tag name.
Numeric ID	`15061`	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 market order volume on the ask side.

2.3.2.6. MARKET_TSE_BidSpecialQuotePrice

The values of the quotation tag **MARKET_TSE_BidSpecialQuotePrice** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Other Values* to detail the special quote price on the bid side:

- 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.

> **Note** A special quote is indicated whenever prices tend to exceed the special quote renewal price intervals. The purpose of a special quote is to prevent major short-term price fluctuations.
>
> Special quotes can be indicated at any time during the trading session, whether before the opening price has been set or during Zaraba trading, if there is any likelihood of inappropriate price fluctuations, for example as a result of a major order imbalance between bids and offers.
>
> Special offer quotes are indicated when the next price is anticipated to be at a price lower than the given appropriate special quote renewal price interval and special bid quotes are indicated when the next price is anticipated to be at a price higher than the special quote renewal price interval.

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

Table 12 MARKET_TSE_BidSpecialQuotePrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_BidSpecialQuotePrice`	FeedOS tag name.
Numeric ID	`15062`	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 special quote price on the bid side.

2.3.2.7. MARKET_TSE_AskSpecialQuotePrice

The values of the quotation tag **MARKET_TSE_AskSpecialQuotePrice** conveyed on the TSE market data stream are disseminated via FeedOS data stream in *Other Values* to detail the special quote price on the ask side:

- 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.

Note	A special quote is indicated whenever prices tend to exceed the special quote renewal price intervals. The purpose of a special quote is to prevent major short-term price fluctuations. Special quotes can be indicated at any time during the trading session, whether before the opening price has been set or during Zaraba trading, if there is any likelihood of inappropriate price fluctuations, for example as a result of a major order imbalance between bids and offers. Special offer quotes are indicated when the next price is anticipated to be at a price lower than the given appropriate special quote renewal price interval and special bid quotes are indicated when the next price is anticipated to be at a price higher than the special quote renewal price interval.

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

Table 13 MARKET_TSE_AskSpecialQuotePrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_TSE_AskSpecialQuotePrice`	FeedOS tag name.
Numeric ID	`15063`	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 special quote price on the ask side.

2.4. MBL and MBO Data*

The MBL book has a 10-level depth and a special limit at the 11th level with a price equal to `UNQUOTED`. This limit represents the aggregation of all the limits beyond the 10th. There is no MBO.

3. Closing Price

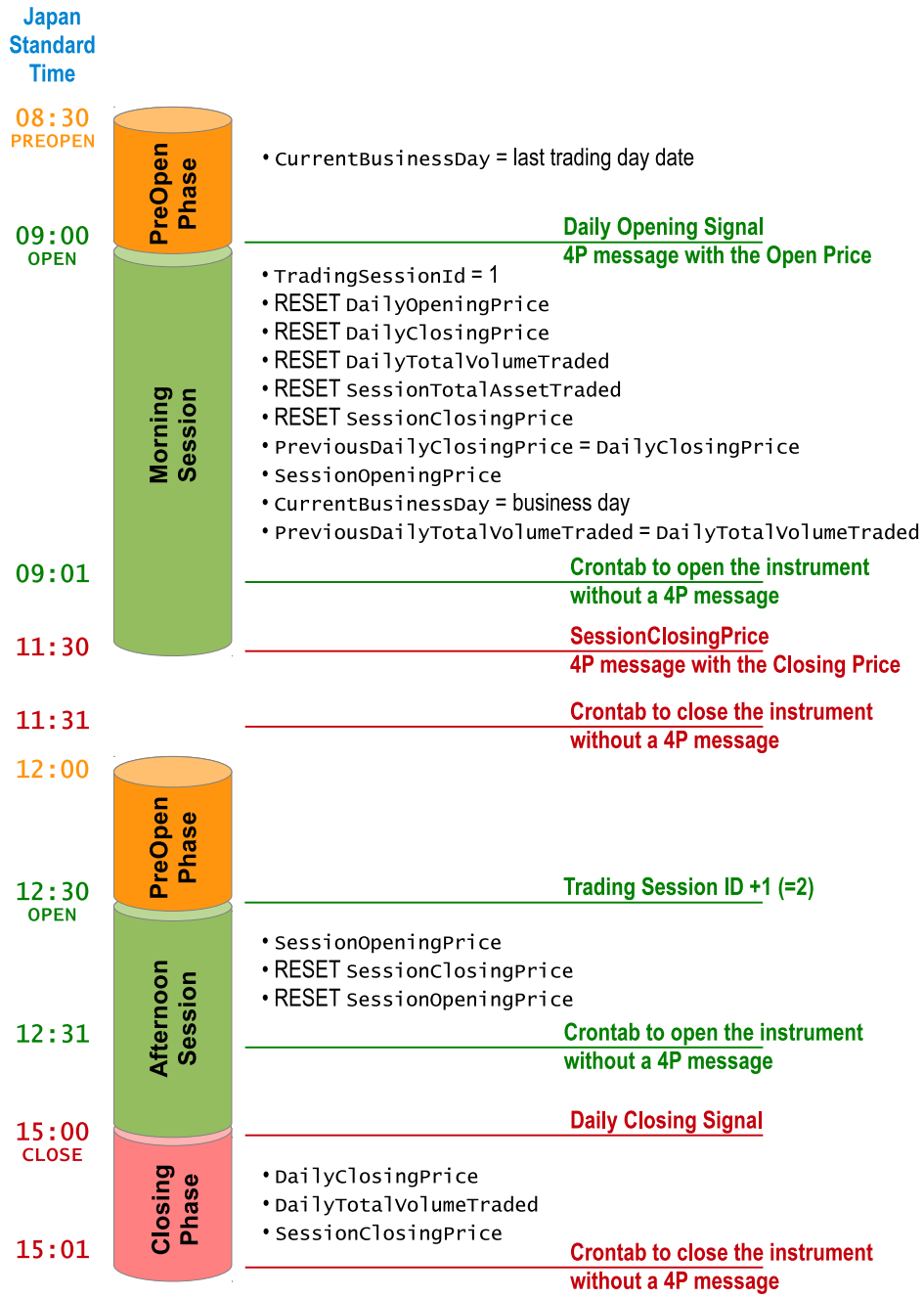
The closing price is the last trade price upon close. There is no settlement price.

4. Multi-Session Kinematics

The following diagram describes the main trading phases and update mechanism on the `XTKS` market:

* 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.

Figure 1 Example of tags update mechanism on the XTKS market during a trading day



For more details about the update mechanism of the fields and tags, and their possible values, see also *FeedOS Quotation Tags Guide*.

The Trading Sessions of XTKS, XFKA, XNGO and XSAP markets are described in the following table:

Table 14 Trading Sessions

Market	TSE/Tokyo AIM				NSE/FSE/SSE			
Trading State	Morning Session		Afternoon Session		Morning Session		Afternoon Session	
	Begin Hour	End Hour	Begin Hour	End Hour	Begin Hour	End Hour	Begin Hour	End Hour
PreOpen (place order only)	08:00:00	09:00:00	12:05:00	12:30:00	08:00:00	09:00:00	12:05:00	12:30:00
Continuous Trading	09:00:00	11:30:00	12:30:00	15:00:00	09:00:00	11:30:00	12:30:00	15:30:00

5. Special Behavior

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

5.1. Level1 Market Data Kinematics – OPEN

In the Level1 Market Data Kinematics **before 2015-06-29**, at 00:00 UTC Time, the exchange sent the OPEN signal and the Trading Status was set to 17=ReadyToTrade, 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   00:00:00:085.762   309284167   *    *    2231    194100    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:085.762   309284167   MARKET_TSE_BidMarketOrderVolume=99300
TE   00:00:00:097.252   309284167   *    *    2231    194800    2231.5    194100
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
SI   00:00:00:103.731   309284167   OPEN    *
TE   00:00:00:103.731   309284167   *    *    *    *    *    *    O
VU   00:00:00:103.731   309284167   TradingSessionId=1    TradingStatus=17
TE   00:00:00:139.616   309284167   *    *    *    *    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:139.616   309284167   MARKET_TSE_BidSpecialQuotePrice=?
MARKET_TSE_AskSpecialQuotePrice=?
TE   00:00:00:142.454   309284167   *    *    2231    195000    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
```

In the Level1 Market Data Kinematics **after 2015-06-29**, the first trade will trigger the OPEN signal and set the Trading Status to `17=ReadyToTrade`, 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   00:00:00:085.762   309284167   *    *    2231    194100    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:085.762   309284167   MARKET_TSE_BidMarketOrderVolume=99300
TE   00:00:00:097.252   309284167   *    *    2231    194800    2231.5    194100
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:139.616   309284167   *    *    *    *    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:139.616   309284167   MARKET_TSE_BidSpecialQuotePrice=?
MARKET_TSE_AskSpecialQuotePrice=?
TE   00:00:00:142.454   309284167   *    *    2231    195000    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:148.938   309284167   *    *    2231    194800    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:156.298   309284167   *    *    *    *    2231.5    194300
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:159.193   309284167   *    *    2231    195000    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:159.193   309284167   MARKET_TSE_BidMarketOrderVolume=99500
MARKET_TSE_AskMarketOrderVolume=145000
TE   00:00:00:163.046   309284167   *    *    2231    194300    2231    194300
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:168.221   309284167   *    *    2231    195000    2231.5    194300
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
TE   00:00:00:238.518   309284167   *    *    2231    195300    *    *
MARKET_TSE_BidQuoteCondition=char{0},MARKET_TSE_AskQuoteCondition=char{0}
VU   00:00:00:238.518   309284167   MARKET_TSE_BidMarketOrderVolume=99800
SI   00:00:12:103.731   309284167   OPEN    *
TE   00:00:12:103.731   309284167   *    *    *    *    *    *    O
VU   00:00:12:103.731   309284167   TradingSessionId=1    TradingStatus=17
TE   00:00:12:103.731   309284167   2231    194300    *    *    *    *    HL
TradeCondition=E=opening_reopening_trade_detail
```

5.2. Daily Prices Management

TSE market data stream does not manage High Speed Index prices. Moreover, the Daily High and Daily Low prices may be wrong after the opening of the afternoon session. However, the Session High and Session Low prices are always correct.

5.3. Market Order Prices

Following the introduction of the `MARKET_TSE_BidMarketOrderVolume` and `MARKET_TSE_AskMarketOrderVolume` tags, the Market Order Prices (AT_BEST Prices) are no longer provided in the Level1 Market Data, as shown in the example below.

Moreover, only these two tags disseminate the information describing the volume of market orders. However, the MBL Level2 does not change, as the markets orders remain in the first position. Subsequently, the Level1 and the first limit of the MBL Level2 may differ.

Level1 Market Data before 2014-04-28

Bid Qty	Bid Price	Ask Price	Ask Qty
1000	*AT_BEST*	*AT_BEST*	**500**

Level2 Market Data before 2014-04-28

Bid Qty	Bid Price	Ask Price	Ask Qty
1000	*AT_BEST*	*AT_BEST*	500
200	**920**	**922**	**300**

Level1 Market Data after 2014-04-28

BidMarketOrderVolume	Bid Qty	Bid Price	Ask Price	Ask Qty	AskMarketOrderVolume
1000	**200**	**920**	**922**	**300**	**500**

Level2 Market Data after 2014-04-28

Bid Qty	Bid Price	Ask Price	Ask Qty
1000	*AT_BEST*	*AT_BEST*	500
200	920	922	300

5.4. ToSTNeT Instrument Reporting

Tokyo Stock Exchange Trading NeTwork (ToSTNeT) trades are managed on dedicated instruments as follows:

- Single Stock and Basket Trades belonging to ToSTNeT-1 are reported on instruments suffixed with `.T1` from 08:20 to 17:30 (Japan Local Time)
- Closing Price Trading belonging to ToSTNeT-2 is reported on instruments suffixed with `.T2` from 08:20 to 17:30 (Japan Local Time)
- Off-Auction Own Share Repurchase Trades belonging to ToSTNeT-3 are reported on instruments suffixed with `.T3` from 08:00 to 08:45 (Japan Local Time) and on every auction during the trading day.

However, S&P Capital IQ Real-Time Solutions has configured the opening and closing of ToSTNeT instruments to include the final reporting of trades as detailed below:

- ToSTNeT-1 between 08:15 and 18:00 (Japan Local Time)
- ToSTNeT-2 between 08:15 and 18:00 (Japan Local Time)
- ToSTNeT-3 between 07:55 and 18:00 (Japan Local Time).

The sample below shows the specific referential data of a ToSTNeT instrument:

```
instr # 147/1011828 = 309293172
    PriceCurrency               string{JPY}
    Symbol                      string{8648}
    SecurityType                string{CS}
    FOSMarketId                 XTKS
    CFICode                     string{ESXXXA}
    RoundLot                    float64{10}
    SecuritySubType             string{11}
    SecurityGroup               string{0113}
    InternalCreationDate        Timestamp{2013-10-22 22:38:12:22
    InternalModificationDate    Timestamp{2015-07-29 09:15:14:65
    InternalSourceId            uint16{40}
    LocalCodeStr                string{8648.T1}
    MBLLayersDesc               string{0}
    OperatingMIC                string{XTKS}
    SegmentMIC                  string{XTK1}
    IndustryCode                string{7050}
```

5.5. Microsecond Timestamp Precision on the Level1 Market Data

The server timestamps 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   309284167    *       *       *       *       41.27   700@2
TE   20:00:48:238.168   309284167    *       *       *       *       47.22   100@1
TE   20:00:48:240.254   309284167    *       *       *       *       48.31   100@1
```

6. 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.