S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

SWX

Reference n°: 20150317 – 22822 – 25850

S&P CAPITAL IQ
McGRAW HILL FINANCIAL

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: SWX
Reference 20150317 – 22822 – 25850
March 19, 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

TABLE OF CONTENTS

FEEDOS™ SWX 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 SWX 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 SWX market data stream, in terms of:

1.1. Available Markets and Branches

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

1.1.1. Markets

The SWX market data stream disseminates informations about the following markets:

Table 1 List of markets available on SWX market data stream

FeedOS Market ID	Market
XSWX	Swiss Exchange
XVTX	VIRT-X
LIQU	Liquidnet System
XQMH	SCOACH Switzerland

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

```
MARKETS
market # 256    CC=CH/SWITZERLAND/ZURICH,DESCR=SWISS EXCHANGE, WEB=www.swx.com
    MIC = XSWX
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 298    CC=GB/UNITED KINGDOM/LONDON,DESCR=VIRT-X, WEB=www.virt-x.com
    MIC = XVTX
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 491    CC=GB/UNITED KINGDOM/LONDON,DESCR=LIQUIDNET SYSTEM, WEB=www.liquidnet.com
    MIC = LIQU
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 498    CC=CH/SWITZERLAND/ZURICH,DESCR=SCOACH SWITZERLAND, WEB=www.scoach.com
    MIC = XQMH
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
```

1.1.2. Branches

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

```
BRANCHES
    { XSWX COFP EMXXXX } qty: 10
    { XSWX CS   ESXXXR } qty: 226
    { XSWX CS   ESXXXX } qty: 536
    { XSWX CS   EXXXXX } qty: 13
    { XSWX GO   DBXXXX } qty: 6265
    { XSWX GO   DCXXXX } qty: 32
    { XSWX MF   EUXXXX } qty: 1560
    { XSWX NONE DBXXXX } qty: 8
    { XSWX NONE DXXXXX } qty: 88
    { XSWX NONE EXXXXB } qty: 54
    { XSWX NONE EXXXXX } qty: 6
    { XSWX NONE MRXXXX } qty: 1
    { XSWX NONE RXXXCX } qty: 5
    { XSWX NONE XXXXXX } qty: 3
    { XVTX COFP EMXXXX } qty: 1
    { XVTX CS   ESXXXR } qty: 29
    { XVTX NONE EMXXXX } qty: 1
    { XVTX NONE EXXXXB } qty: 2
    { LIQU CS   ESXXXX } qty: 3798
    { LIQU NONE XXXXXX } qty: 749
    { LIQU PS   EPXXXX } qty: 104
    { XQMH NONE MRXXXX } qty: 22583
    { XQMH WAR  RWCXCX } qty: 299
    { XQMH WAR  RWCXPX } qty: 352
    { XQMH WAR  RWTXCX } qty: 424
    { XQMH WAR  RWTXPX } qty: 319
    { XQMH WAR  RWXXCX } qty: 41949
    { XQMH WAR  RWXXPX } qty: 21209
```

1.2. Types of Instruments

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

1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 256/512845 = 537383757
    PriceCurrency               string{CHF}
    Symbol                      string{ABBNE}
    Issuer                      string{ABB Ltd}
    Description                 string{ABB LTD N 2. LINIE}
    SecurityType                string{CS}
    FOSMarketId                 XSWX
    PriceType                   uint8{2}
    CFICode                     string{ESXXXR}
    RoundLot                    float64{1}
    MinTradeVol                 float64{0}
    SecuritySubType             string{Registered Share}
    DatedDate                   Timestamp{2014-09-16}
    SecurityGroup               string{2110}
    MarketSegmentID             string{597}
    MarketSegmentDesc           string{Separate Trading Lines}
    InternalCreationDate        Timestamp{2014-09-15 00:00:05:405}
    InternalModificationDate    Timestamp{2015-03-04 14:44:35:357}
    InternalSourceId            uint16{29}
    InternalEntitlementId       SWX
    InternalMagic               string{Mid & Small Cap Shares}
    LocalCodeStr                string{CH0253301128_CHF}
    ISIN                        string{CH0253301128}
    Telekurs_Valor              string{25330112}
    PriceIncrement_dynamic_TableId  uint32{3342436}
    SecurityTradingId           string{3232940}
    OperatingMIC                string{XSWX}
    CCP_Eligible                bool{False}
    MARKET_SWX_IssuerCountry    string{CH}
    MARKET_SWX_TradingSessionID string{ABdI}
    MARKET_SWX_ListingStateCode string{LI}
    MARKET_SWX_ListingStateDesc string{Listed}
```

1.2.2. Bonds

The sample below illustrates the details of a bond:

```
instr # 256/511422 = 537382334
    PriceCurrency               string{CHF}
    Symbol                      string{PB581}
    Issuer                      string{Pfandbriefbank}
    Description                 string{1.625 PB S581 13-30}
    SecurityType                string{GO}
    StdMaturity                 string{203007}
    FOSMarketId                 XSWX
    CouponRate                  float64{1.625}
    PriceType                   uint8{1}
    CFICode                     string{DBXXXX}
    RoundLot                    float64{5000}
    MinTradeVol                 float64{0}
    SecuritySubType             string{Swiss Pfandbriefe}
    DatedDate                   Timestamp{2013-04-25}
    SecurityGroup               string{2210}
    MarketSegmentID             string{590}
    MarketSegmentDesc           string{Bonds - CHF - Domestic and Foreign}
    InternalCreationDate        Timestamp{2015-03-17 12:44:46:599}
    InternalModificationDate    Timestamp{2015-03-17 12:44:46:599}
    InternalSourceId            uint16{29}
    InternalEntitlementId       int32{1093}
    InternalMagic               string{Bonds - CHF}
    LocalCodeStr                string{CH0211588949_CHF}
    ISIN                        string{CH0211588949}
    MaturityYear                uint16{2030}
    MaturityMonth               uint8{7}
    MaturityDay                 uint8{3}
    Telekurs_Valor              string{21158894}
    PriceIncrement_dynamic_TableId      uint32{3342439}
    SecurityTradingId           string{3171739}
    OperatingMIC                string{XSWX}
    CCP_Eligible                bool{True}
    DynamicVariationRange       float64{3}
    MARKET_SWX_IssuerCountry    string{CH}
    MARKET_SWX_TradingSessionID string{BBdB}
    MARKET_SWX_ListingStateCode string{LI}
    MARKET_SWX_ListingStateDesc string{Listed}
```

1.2.3. Warrants

The sample below illustrates the details of a warrant:

```
instr # 498/770265 = 1045151961
    PriceCurrency               string{CHF}
    Symbol                      string{VTGBAP}
    Issuer                      string{Bank Vontobel AG}
    Description                 string{VTGBAP VON C 09/15}
    SecurityType                string{WAR}
    StdMaturity                 string{20150909}
    StrikePrice                 float64{1.55}
    FOSMarketId                 XQMH
    Factor                      float64{10}
    ContractMultiplier          float64{1}
    PriceType                   uint8{2}
    CFICode                     string{RWCXCX}
    RoundLot                    float64{1}
    MinTradeVol                 float64{0}
    SecuritySubType             string{Warrant on Currency}
    DatedDate                   Timestamp{2014-11-27}
    StrikeCurrency              string{USD}
    SecurityGroup               string{3130}
    MarketSegmentID             string{580}
    MarketSegmentDesc           string{SIX Structured Products}
    InternalCreationDate        Timestamp{2015-03-17 12:44:46:551}
    InternalModificationDate    Timestamp{2015-03-17 12:44:46:551}
    InternalSourceId            uint16{29}
    InternalEntitlementId       int32{10090}
    InternalMagic               string{Structured Products Exchange - Warrants}
    LocalCodeStr                string{CH0256317592_CHF}
    ISIN                        string{CH0256317592}
    UnderlyingLocalCodeStr      string{ZZGBPUSD0005}
    MaturityYear                uint16{2015}
    MaturityMonth               uint8{9}
    MaturityDay                 uint8{18}
    Telekurs_Valor              string{25631759}
    PriceIncrement_dynamic_TableId      uint32{3342440}
    SecurityTradingId           string{3243064}
    OperatingMIC                string{XQMH}
    CCP_Eligible                bool{False}
    MARKET_SWX_IssuerCountry    string{CH}
    MARKET_SWX_TradingSessionID string{DJU}
    MARKET_SWX_ListingStateCode string{LI}
    MARKET_SWX_ListingStateDesc string{Listed}
```

1.2.4. Miscellaneous

The sample below illustrates the details of a miscellaneous instrument:

```
instr # 498/761805 = 1045143501
    PriceCurrency               string{EUR}
    Symbol                      string{VONLRE}
    Issuer                      string{VFP Dubai}
    Description                 string{4.69767 VON/PAH3 15}
    SecurityType                string{NONE}
    StdMaturity                 string{201510}
    StrikePrice                 float64{68.14}
    FOSMarketId                 XQMH
    CouponRate                  float64{4.69767}
    Factor                      float64{14.67567}
    ContractMultiplier          float64{1}
    PriceType                   uint8{1}
    CFICode                     string{MRXXXX}
    RoundLot                    float64{1000}
    MinTradeVol                 float64{0}
    SecuritySubType             string{Structured Products Bonds}
    DatedDate                   Timestamp{2014-09-25}
    StrikeCurrency              string{EUR}
    SecurityGroup               string{3131}
    MarketSegmentID             string{580}
    MarketSegmentDesc           string{SIX Structured Products}
    InternalCreationDate        Timestamp{2015-03-17 12:44:44:990}
    InternalModificationDate    Timestamp{2015-03-17 12:44:44:990}
    InternalSourceId            uint16{29}
    InternalEntitlementId       int32{1090}
    InternalMagic               string{Structured Products Exchange - Structured Products}
    LocalCodeStr                string{CH0244612245_EUR}
    ISIN                        string{CH0244612245}
    UnderlyingLocalCodeStr      string{DE000PAH0038}
    MaturityYear                uint16{2015}
    MaturityMonth               uint8{10}
    MaturityDay                 uint8{23}
    Telekurs_Valor              string{24461224}
    PriceIncrement_dynamic_TableId      uint32{3342439}
    SecurityTradingId           string{3233730}
    OperatingMIC                string{XQMH}
    CCP_Eligible                bool{False}
    MARKET_SWX_IssuerCountry    string{AE}
    MARKET_SWX_TradingSessionID string{DJU}
    MARKET_SWX_ListingStateCode string{LI}
    MARKET_SWX_ListingStateDesc string{Listed}
```

1.3. Specific Referential Tags

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

1.3.1. SecurityGroup

The values of the referential tag **SecurityGroup** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to identify the group to which a financial instrument belongs.

FeedOS implementation of the values currently available for the tag `SecurityGroup` is described below:

Table 2 SecurityGroup – technical implementation in FeedOS

Component	Value	Description
Tag Name	`SecurityGroup`	FeedOS tag name.
Numeric ID	`1151`	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 retrieved from`* `billingSegmentCode` *`and`* `billingSegmentDesc`*`]`*	An ***exchange specific value***, identifying the group to which a financial instrument belongs.
Possible Values	`1110`	Blue Chip Shares
	`2110`	Mid & Small Cap Shares
	`2120`	Secondary Listed & Misc. Shares
	`2210`	CHF Bonds
	`2220`	International Bonds
	`2310`	ETF
	`2320`	ETSF
	`2330`	Investment Funds
	`2340`	Sponsored Funds
	`2410`	ETP
	`3111`	Leverage Products on Scoach
	`3121`	Participation Products on Scoach
	`3122`	Yield enhancements Products on Scoach
	`3123`	Capital protection Products on Scoach
	`3124`	Miscellaneous Derivatives on Scoach

1.3.2. MarketSegmentID and MarketSegmentDesc

The values of the referential tags **MarketSegmentID** and **MarketSegmentDesc** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Referential* to detail the ID of the market segment and its description.

FeedOS implementation of the tags `MarketSegmentID` and `MarketSegmentDesc` is described below:

Table 3 MarketSegmentID and MarketSegmentDesc – technical implementation in QuantFEED®

Component	Value		Description
Tag Name	`MarketSegmentID`	`MarketSegmentDesc`	FeedOS tag name.
Numeric ID	1300	1396	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***, detailing the ID of the market segment and its description.
Possible Values	<EMPTY>	<EMPTY>	
	AF	Funds and ETF	
	HS	Main Market	
	IA	International Bonds	
	IG	Real Estate	
	IV	Investment Companies	
	LC	Local Caps	
	SP	SWX Sponsored Segment	
	26	Blue Chip Shares	
	580	SIX Structured Products	
	581	International Bonds	
	582	International Bonds Min Denom	
	583	International Bonds Convertible	
	584	ETF	
	585	ETF on Swiss Confederation Bonds	
	588	ETP	
	589	Swiss Confederation Bonds CHF	
	590	Bonds CHF	
	591	Mid-/Small-Cap Shares	
	592	Secondary Listing Shares	
	594	Investment Funds	
	596	Convertible and Warrant Bonds CHF	
	597	Separate Trading Lines	
	612	Sponsored Funds	
	613	Sponsored Foreign Shares	

1.3.3. InternalModificationDate

The values of the referential tag **InternalModificationDate** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Referential* to specify the date when the referential data of an instrument has changed internally.

FeedOS implementation of the values available for the tag `InternalModificationDate` is described below:

Table 4 InternalModificationDate – technical implementation in FeedOS

Component	Value	Description
Tag Name	`InternalModificationDate`	FeedOS tag name.
Numeric ID	`9401`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Timestamp`	Timestamp data type.
Format / Possible Values	*`[Internal Specific Value]`*	An ***internal specific value***, detailing the date when the referential data of an instrument has changed internally. **NOTE:** After 2014-07-07, the update mechanism of the tag `InternalModificationDate` changes. Thus, the timestamp will no longer be updated on a daily basis, unless there is a significant change in the referential data of the instrument.

1.3.4. UnderlyingLocalCode

The values of the referential tag **UnderlyingLocalCode** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to detail the local code.

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

Table 5 UnderlyingLocalCode – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`UnderlyingLocalCode`	FeedOS tag name.
Numeric ID	`9510`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the underlying local code.

1.3.5. Telekurs_Valor

The values of the referential tag **Telekurs_Valor** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to identify the Telekurs securities.

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

Table 6 Telekurs_Valor – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`Telekurs_Valor`	FeedOS tag name.
Numeric ID	`9521`	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***, indicating the Telekurs securities.

1.3.6. SecurityTradingId

The values of the referential tag **SecurityTradingId** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Referential* to specify the trading ID of a security.

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

Table 7 SecurityTradingId – technical implementation in FeedOS

Component	Value	Description
Tag Name	`SecurityTradingId`	FeedOS tag name.
Numeric ID	`9525`	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***, specifying trading ID of a security.

1.3.7. OperatingMIC and SegmentMIC

The values of the referential tags **OperatingMIC** and **SegmentMIC** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to reflect SWX adoption of the ISO 10383:2012 standard. This new edition of the ISO standard refines the level of granularity on SWX 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 table below:

Table 8 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	`LIQU`	`LIQU`	LIQUIDNET SYSTEMS
	`XQMH`	`XQMH`	SCOACH SWITZERLAND
	`XSWX`	`XSWX`	SWISS EXCHANGE
	`XSWX`	`XVTX`	SIX SWISS EXCHANGE AG

1.3.8. CCP_Eligible

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

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

Table 9 CCP_Eligible – technical implementation in FeedOS

Component	Value	Description
Tag Name	`CCP_Eligible`	FeedOS tag name.
Numeric ID	`9552`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Bool`	Bool data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing whether an instrument is cleared via the CCP.
Possible Values	`True`	CCP eligibility and post trade anonymity.
	`False`	Default value, not sent.

1.3.9. DynamicVariationRange

The values of the referential tag **DynamicVariationRange** conveyed on SWX 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.

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

Table 10 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, as shown in the following example.

1.3.10. MARKET_SWX_IssuerCountry

The values of the referential tag **MARKET_SWX_IssuerCountry** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to uniquely identify the incorporation country of the instrument issuer.

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

Table 11 MARKET_SWX_IssuerCountry – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_SWX_IssuerCountry`	FeedOS tag name.
Numeric ID	`11350`	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***, uniquely identifying the incorporation country of the instrument's issuer.

1.3.11. MARKET_SWX_TradingSessionID

The values of the referential tag **MARKET_SWX_TradingSessionID** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to detail the unique identifier of a Trading Session applied to a Traded Instrument Order Book.

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

Table 12 MARKET_SWX_TradingSessionID – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_SWX_TradingSessionID`	FeedOS tag name.
Numeric ID	`11353`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the unique identifier of a Trading Session applied to a Traded Instrument Order Book.

1.3.12. MARKET_SWX_ListingStateCode

The values of the referential tag **Listing State Code** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to detail the instrument status which implies the rule book governing its trading.

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

Table 13 MARKET_SWX_ListingStateCode – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_SWX_ListingStateCode`	FeedOS tag name.
Numeric ID	`11354`	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 instrument status which implies the rule book governing its trading.

Table 13 MARKET_SWX_ListingStateCode – technical implementation in QuantFEED® (Continued)

Component	Value	Description
Possible Values	DK	Delisted
	LI	Listed
	NK	Not-listed
	PZ	Provisional Listing

1.3.13. MARKET_SWX_ListingStateDesc

The values of the referential tag **MARKET_SWX_ListingStateDesc** conveyed on SWX market data stream are disseminated via FeedOS data stream in *Referential* to detail the instruments state code description.

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

Table 14 MARKET_SWX_ListingStateDesc – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_SWX_ListingStateDesc`	FeedOS tag name.
Numeric ID	`11355`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the instruments state code description.

2. Quotation Data

The following sections describe the characteristics of the quotation data on the SWX 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 the SWX market data stream:

```
InstrumentStatusL1
-- 498/771414
    BID: 0.13   250000  @1
    ASK: 0.14   250000  @1
        LastPrice                   float64{0.15}
        LastTradeQty                float64{230000}
        DailyTotalVolumeTraded      float64{0}
        DailyTotalAssetTraded       float64{0}
        LastTradePrice              float64{0.15}
        LastTradeTimestamp          Timestamp{2015-03-12 12:33:02:308}
        InternalDailyOpenTimestamp  Timestamp{2015-03-17 12:44:57:087}
        InternalDailyCloseTimestamp Timestamp{2015-03-16 16:15:00:077}
        InternalDailyHighTimestamp  Timestamp{2015-03-12 12:33:02:314}
        InternalDailyLowTimestamp   Timestamp{2015-03-12 12:33:02:314}
        InternalPriceActivityTimestamp  Timestamp{2015-03-17 12:44:54:073}
        TradingStatus               17=ReadyToTrade
        PreviousDailyTotalVolumeTraded  float64{230000}
        PreviousDailyTotalAssetTraded   float64{34500}
        PreviousDailyClosingPrice   float64{0.15}
        PreviousBusinessDay         Timestamp{2015-03-12}
        CurrentBusinessDay          Timestamp{2015-03-17}
        InternalDailyClosingPriceType   char{a}
        PriceActivityMarketTimestamp    Timestamp{2015-03-17 12:44:54:073}
        TradingReferencePrice       float64{0.14}
        MARKET_SWX_BookCondition    int32{3}
        MARKET_SWX_SecurityTradingStatus    int32{17}
        MARKET_SWX_TradingSessionSubID  string{2}
```

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 SWX 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 15 TradingStatus – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingStatus`	FeedOS tag name.
Numeric ID	`9100`	FeedOS unique ID disseminated on 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
	`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 SWX 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 SWX market data stream:

- 2.3.1.1. TradeCondition
- 2.3.1.2. MARKET_SWX_TradeTypeIndicator
- 2.3.1.3. MARKET_SWX_LastAuctionQty
- 2.3.1.4. MARKET_SWX_TradeOffExchangeFlag
- 2.3.1.5. MARKET_SWX_TradingPhase
- 2.3.1.6. TradeID (Optional).

2.3.1.1. TradeCondition

Each time an on-book trade with 'InternalCross' type occurs, the values of the quotation tag **TradeCondition** conveyed on the SWX 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 16 TradeCondition – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradeCondition`	FeedOS tag name.
Numeric ID	`277`	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 particular condition applicable to the trade.
Possible Values	`X`	Crossed

2.3.1.2. MARKET_SWX_TradeTypeIndicator

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

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

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

Table 17 MARKET_SWX_TradeTypeIndicator – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_TradeTypeIndicator`	FeedOS tag name.
Numeric ID	`15450`	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 trade type.
Possible Values	`Empty or Space`	Default value, not sent.
	`30`	Special Price (FIX standard value)

2.3.1.3. MARKET_SWX_LastAuctionQty

Each time a trade occurs, the values of the quotation tag **MARKET_SWX_LastAuctionQty** conveyed on the SWX 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 `MARKET_SWX_LastAuctionQty` is described in the table below:

Table 18 MARKET_SWX_LastAuctionQty – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_LastAuctionQty`	FeedOS tag name.
Numeric ID	`15451`	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	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, indicating the quantity of the last auction.

2.3.1.4. MARKET_SWX_TradeOffExchangeFlag

The values of the quotation tag **MARKET_SWX_TradeOffExchangeFlag** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Context* to detail the status of an 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_SWX_TradeOffExchangeFlag` is described in the table below:

Table 19 MARKET_SWX_TradeOffExchangeFlag – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_TradeOffExchangeFlag`	FeedOS tag name.
Numeric ID	`15452`	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 status of an instrument.
Possible Values	Y	Off-Exchange Instrument
	N	Regular Instrument

2.3.1.5. MARKET_SWX_TradingPhase

The values of the quotation tag **MARKET_SWX_TradingPhase** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Context* to detail the origin of the trades generated from an auction phase:

- 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_SWX_TradingPhase` is described in the table below:

Table 20 MARKET_SWX_TradingPhase – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_TradingPhase`	FeedOS tag name.
Numeric ID	`15453`	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 origin of the trades generated from an auction phase.
Possible Values	`0`	Trading
	`1`	Auction
	`2`	First Auction
	`3`	Last Auction

2.3.1.6. TradeID (Optional)

Each time a trade occurs, the values of the quotation context tag **TradeID** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Context* only for S&P Capital IQ Real-Time Solutions customers using a dedicated SWX feed handler 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 21 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. **Available upon request only for S&P Capital IQ Real-Time Solutions customers using a dedicated SWX feed handler.**

2.3.2. Other Values

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

- 2.3.2.1. InternalDailyClosingPriceType
- 2.3.2.2. TradingReferencePrice
- 2.3.2.3. MARKET_SWX_BookCondition
- 2.3.2.4. MARKET_SWX_SecurityTradingStatus

- 2.3.2.5. MARKET_SWX_TradingSessionSubID.

2.3.2.1. InternalDailyClosingPriceType

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

Table 22 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. TradingReferencePrice

The values of the quotation tag **TradingReferencePrice** are disseminated via FeedOS data stream in *Other Values* to indicate the superior price limit:

- 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 `TradingReferencePrice` is described in the following table:

Table 23 TradingReferencePrice – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingReferencePrice`	FeedOS tag name.
Numeric ID	`9370`	FeedOS unique ID disseminated on 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 superior price limit.

Caution	The tag `TradingReferencePrice` is updated after the market closes, but not during the trading phase.

2.3.2.3. MARKET_SWX_BookCondition

The values of the quotation tag **MARKET_SWX_BookCondition** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Other Values* to indicate a particular condition that affects the Book:

- 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_SWX_BookCondition` is described in the table below:

Table 24 MARKET_SWX_BookCondition – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_BookCondition`	FeedOS tag name.
Numeric ID	`14452`	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	`Int32`	Int32 data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, indicating a particular condition that affects the Book.
Possible Values	`0`	Delayed Opening
	`1`	Delayed Opening with Non Opening
	`2`	Non Opening
	`3`	None
	`4`	Stop Trading
	`5`	Stop Trading with Non Opening
	`6`	Underlying Condition
	`7`	Underlying Condition with Non Opening

2.3.2.4. MARKET_SWX_SecurityTradingStatus

Each time a modification of the security status occurs, the values of the quotation tag **MARKET_SWX_SecurityTradingStatus** conveyed on the SWX 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_SWX_SecurityTradingStatus` is described in the table below:

Table 25 MARKET_SWX_SecurityTradingStatus – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_SecurityTradingStatus`	FeedOS tag name.
Numeric ID	`14453`	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	`Int32`	Int32 data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the current status of a security.
Possible Values	2	Trading Halt
	17	Ready to Trade

2.3.2.5. MARKET_SWX_TradingSessionSubID

The values of the quotation tag **MARKET_SWX_TradingSessionSubID** conveyed on the SWX market data stream are disseminated via FeedOS data stream in *Other Values* to detail the trading schedule transition:

- 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_SWX_TradingSessionSubID` is described in the table below:

Table 26 MARKET_SWX_TradingSessionSubID – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_SWX_TradingSessionSubID`	FeedOS tag name.
Numeric ID	`14454`	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 trading schedule transition.

Table 26 MARKET_SWX_TradingSessionSubID – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	0	Start of Day
	1	Pre-Open
	2	Open
	3	Holiday
	4	Accepting
	5	Break
	6	End Break
	7	Close
	8	Auction
	9	Closing Auction
	A	Auction Close
	B	End of Day

2.4. MBL, MBO and BBO Data*

The MBL book has a 5-level depth. The MBO book is full depth.

3. Official Closing Price

On the market SWX, the closing price is provided by the market. If it is not sent by the market, the last trade is used instead. 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 SWX market data stream in terms of:

- 4.1. End of Auction Kinematics
- 4.2. CLOSE Kinematics
- 4.3. Microsecond Timestamp Precision on the Level1 Market Data.

* 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.

4.1. End of Auction Kinematics

In the kinematics before 2015-03-30, the `LastAuctionPrice` and `LastAuctionVolume` were reset at the end of the Auction Phase, 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..."

VU   07:57:17:933  537371421   LastAuctionPrice=40.25  LastAuctionVolume=2815
VU   07:59:54:435  537371421   LastAuctionVolume=3077
VU   07:59:59:521  537371421   LastAuctionVolume=3148
SI   08:01:01:016  537371421   OPEN    *
TE   08:01:01:016  537371421   *       *       *       *       *       *       O
VU   08:01:01:016  537371421   MARKET_SWX_TradingSessionSubID=2        TradingStatus=17
TE   08:01:01:040  537371421   *       *       40.25   554@1   40.3    150@1
VU   08:01:01:040  537371421   LastAuctionPrice=?      LastAuctionVolume=?
TE   08:01:01:000  537371421   40.25   150     *       *       *       *       HL
MARKET_SWX_TradingPhase=2
VU   08:01:01:000  537371421   DailyOpeningPrice=40.25
TE   08:01:01:000  537371421   40.25   100     *       *       *       *
MARKET_SWX_TradingPhase=2
TE   08:01:01:000  537371421   40.25   2       *       *       *       *
MARKET_SWX_TradingPhase=2
TE   08:01:01:000  537371421   40.25   22      *       *       *       *
MARKET_SWX_TradingPhase=2
```

In the kinematics after 2015-03-30, the `LastAuctionPrice` and `LastAuctionVolume` will be reset at the end of the Auction Phase. The `LastAuctionPrice` will be resent with a new value, when provided by the exchange. Moreover, the `LastAuctionPrice` will be available in the snapshot during the trading day, but the `LastAuctionVolume` will be empty, 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..."

VU   07:57:17:933.323  537371421   LastAuctionPrice=40.25  LastAuctionVolume=2815
VU   07:59:54:435.525  537371421   LastAuctionVolume=3077
VU   07:59:59:521.728  537371421   LastAuctionVolume=3148
SI   08:01:01:016.121  537371421   OPEN    *
TE   08:01:01:016.123  537371421   *       *       *       *       *       *       O
VU   08:01:01:016.180  537371421   MARKET_SWX_TradingSessionSubID=2        TradingStatus=17
TE   08:01:01:040.254  537371421   *       *       40.25   554@1   40.3    150@1
VU   08:01:01:040.287  537371421   LastAuctionPrice=?      LastAuctionVolume=?
TE   08:01:01:000.321  537371421   40.25   150     *       *       *       *       HL
MARKET_SWX_TradingPhase=2
VU   08:01:01:000.457  537371421   DailyOpeningPrice=40.25 LastAuctionPrice=40.25
TE   08:01:01:000.501  537371421   40.25   100     *       *       *       *
MARKET_SWX_TradingPhase=2
TE   08:01:01:000.578  537371421   40.25   2       *       *       *       *
MARKET_SWX_TradingPhase=2
TE   08:01:01:000.648  537371421   40.25   22      *       *       *       *
MARKET_SWX_TradingPhase=2
```

4.2. CLOSE Kinematics

If trades occur during the trading day, the market sends the CLOSE signal and the closing price. However, if no trade occurs during the trading day, the market sends only the CLOSE signal, without the closing price, as shown in the examples below:

Sample SWX CLOSE kinematics for a traded instrument

```
04:50:11:327  *       *       !       0       !       0
04:50:11:334  *       *       !       0       !       0
04:50:11:341  *       *       !       0       !       0
04:50:11:352  *       *       !       0       !       0
04:50:12:257  *       *       !       0       !       0
04:50:12:266  *       *       !       0       !       0
04:55:00:662  *       *       91      40@1    91.8    935@2
04:55:00:894  MARKET_SWX_TradingSessionSubID=0
04:55:00:894  MARKET_SWX_TradingSessionSubID=1        TradingStatus=21
08:00:25:021  OPEN    *
08:00:25:021  *       *       *       *       *       *       O
08:00:25:021  MARKET_SWX_TradingSessionSubID=2        TradingStatus=17
11:41:26:004  *       *       91.25   150@1   *       *
14:47:33:884  *       *       91      40@1    *       *
14:47:33:879  91.25   150     *       *       *       *       HL
14:47:33:879  DailyOpeningPrice=91.25
15:12:26:324  *       *       91      55@2    *       *
16:20:00:013  MARKET_SWX_TradingSessionSubID=9        TradingStatus=5
16:20:00:022  *       *       91      40@1    *       *
16:28:02:671  *       *       AT_BEST 1@1     *       *
16:28:02:671  LastAuctionPrice=91.8   LastAuctionVolume=1
16:31:10:107  CLOSE   91.25
16:31:10:107  91.25   *       *       *       *       *       C
16:31:10:107  MARKET_SWX_TradingSessionSubID=A        TradingStatus=18
16:31:09:999  91.8    1       *       *       *       *               MARKET_SWX_TradingPhase=3
16:31:10:154  *       *       91      40@1    91.8    934@2
16:31:10:154  LastAuctionPrice=?      LastAuctionVolume=?
16:35:09:918  DailyClosingPrice=91.8  DailyHighPrice=91.8     DailyTotalAssetTraded=13779.3
16:45:40:971  9370=91.8
21:00:00:018  MARKET_SWX_TradingSessionSubID=B
```

Sample SWX CLOSE kinematics for a not-traded instrument

```
04:50:11:003  *       *       !       0       !       0
04:50:11:232  *       *       !       0       !       0
04:55:00:905  MARKET_SWX_TradingSessionSubID=0
04:55:00:909  MARKET_SWX_TradingSessionSubID=1        TradingStatus=21
04:55:06:242  *       *       !       0       !       0
08:31:42:013  OPEN    *
08:31:42:013  *       *       *       *       *       *       O
08:31:42:013  MARKET_SWX_TradingSessionSubID=2        TradingStatus=17
08:35:14:056  *       *       101.09  100000@1        *       *
08:35:24:984  *       *       *       *       101.84  100000@1
16:00:00:029  CLOSE   *
16:00:00:029  *       *       *       *       *       *       C
16:00:00:029  MARKET_SWX_TradingSessionSubID=7        TradingStatus=18
16:00:00:070  *       *       !       0       !       0
21:00:00:058  MARKET_SWX_TradingSessionSubID=B
```

4.3. Microsecond Timestamp Precision on the Level1 Market Data

Effective 2015-03-30, the server timestamps will display microsecond units on the Level1 Market Data, as shown in the example below (highlighted in green):

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

TE   15:01:01:000.578  537371421   40.25   18      *       *       *       *
TE   15:02:06:028.392  537371421   40.25   52      *       *       *       *
```

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.