



S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

RTS FAST

Reference n°: 20150814 – 23930 – 28265 – 28266

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: RTS FAST
Reference 20150814 – 23930 – 28265 – 28266
August 14, 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

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

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

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

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

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

TABLE OF CONTENTS

FeedOS™ RTS FAST Feed Description	1
1. Referential Data	1
1.1. Available Markets and Branches	1
1.1.1. Markets	1
1.1.2. Branches	2
1.2. Types of Instruments	2
1.2.1. Equities	3
1.2.2. Futures	3
1.2.3. Indices	4
1.2.4. Options	4
1.2.5. Bonds	5
1.2.6. Multilegs	5
1.2.7. Commodities	6
1.3. Specific Referential Tags	6
1.3.1. StdMaturity	6
1.3.2. ContractMultiplier	7
1.3.3. MarketSegmentID	7
1.3.4. OperatingMIC	7
2. Quotation Data	8
2.1. Quotation Values	9
2.2. TradingStatus	10
2.3. Specific Quotation Tags	10
2.3.1. Trade Conditions	10
2.3.1.1. TradeID	10
2.3.2. Other Values	11
2.3.2.1. LowLimitPrice	11
2.3.2.2. HighLimitPrice	12
2.3.2.3. TradingSessionId	12
2.3.2.4. SessionTotalOffBookAssetTraded	13
2.3.2.5. SessionTotalOffBookVolumeTraded	13
2.3.2.6. SessionTotalVolumeTraded	14
2.3.2.7. PreviousSessionClosingPrice	14
2.3.2.8. SessionHighPrice	15
2.3.2.9. SessionLowPrice	15
2.3.2.10. SessionVWAPPrice	16
2.3.2.11. SessionTotalAssetTraded	16
2.3.2.12. DailySettlementPrice	17
2.3.2.13. CurrentBusinessDay	17
2.3.2.14. OpenInterest	18
2.3.2.15. InternalDailyClosingPriceType	18
2.3.2.16. PriceActivityMarketTimestamp	19

2.3.2.17. SettlementPriceDate	20
2.4. MBL and MBO Data	20
3. Closing Price	20
4. Special Behavior	20
5. Finding the Latest Information	21



FEEDOS™ RTS FAST 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 RTS FAST market data stream, their possible values and current FeedOS technical implementation.

The topics this feed description covers include:

- [1. Referential Data](#)
- [2. Quotation Data](#)
- [3. Closing Price](#)
- [4. Special Behavior](#)
- [5. Finding the Latest Information.](#)

1. Referential Data

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

- [1.1. Available Markets and Branches](#)
- [1.2. Types of Instruments](#)
- [1.3. Specific Referential Tags.](#)

1.1. Available Markets and Branches

This section details the list of [Markets](#) and [Branches](#) available on the RTS FAST market data stream.

1.1.1. Markets

The RTS FAST market data stream disseminates informations about the following markets:

Table 1 List of markets available on the RTS FAST market data stream

FeedOS Market ID	Market
RTSX	RTS – Russian Stock Exchange
UKEX	Ukrainian Exchange
ETSC	ETS Eurasian Trading System Commodity Exchange

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

```
MARKETS
market # 209      CC=RU/RUSSIA/MOSCOW,DESCR=RTS STOCK EXCHANGE,WEB=www.rtsnet.ru
  MIC = RTSX
  TimeZone = Europe/Moscow
  Country = RU
  NbMaxInstruments = 2000000
market # 372      CC=UA/UKRAINE/KIEV,DESCR=UKRAINIAN EXCHANGE,WEB=www.ux.ua,OLD=ECAG,SEQNUM=1
  MIC = UKEX
  TimeZone = Europe/Kiev
  Country = UA
  NbMaxInstruments = 2000000
market # 489      CC=KZ/KAZAKHSTAN/ALMATY,DESCR=ETS EURASIAN TRADING SYSTEM COMMODITY
EXCHANGE,WEB=www.ets.kz/en/,OLD=FRAA,SEQNUM=1
  MIC = ETSC
  TimeZone = Asia/Almaty
  Country = KZ
  NbMaxInstruments = 2000000
```

1.1.2. Branches

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

```
BRANCHES
{ RTSX CS ESXXXX } qty: 1333
{ RTSX FUT FXXXXX } qty: 266
{ RTSX INDEX TIXXXX } qty: 1106
{ RTSX MLEG FXXXXX } qty: 35
{ RTSX OPT OCAFPX } qty: 10058
{ RTSX OPT OPAFPX } qty: 10058
{ UKEX BON DBXXXX } qty: 2
{ UKEX CS DBXXXX } qty: 39
{ UKEX CS ESXXXX } qty: 421
{ UKEX FUT FXXXXX } qty: 15
{ UKEX INDEX TIXXXX } qty: 62
{ UKEX OPT OCAFPX } qty: 552
{ UKEX OPT OPAFPX } qty: 552
{ UKEX TB DBXXXX } qty: 106
{ ETSC COMMODITY TTAXXX } qty: 168
{ ETSC COMMODITY TTETXXX } qty: 748
```

1.2. Types of Instruments

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

- [1.2.1. Equities](#)
- [1.2.2. Futures](#)
- [1.2.3. Indices](#)

- [1.2.4. Options](#)
- [1.2.5. Bonds](#)
- [1.2.6. Multilegs](#)
- [1.2.7. Commodities.](#)

1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 209/1018941 = 439323709
  PriceCurrency      string{USD}
  Symbol             string{mpch}
  Description         string{ОАО ММП имени В.В. Чернышева, ао}
  SecurityType       string{CS}
  FOSMarketId        RTSX
  ContractMultiplier float64{1}
  CFICode            string{ESXXXX}
  MarketSegmentID    string{SBOARD}
  InternalCreationDate Timestamp{2015-05-10 20:01:00:733}
  InternalModificationDate Timestamp{2015-05-10 20:01:00:733}
  InternalSourceId    uint16{55}
  InternalEntitlementId int32{1176}
  LocalCodeStr        string{1229395}
  PriceIncrement_static float64{0.0001}
  OperatingMIC         string{RTSX}
```

1.2.2. Futures

The sample below illustrates the details of a future:

```
instr # 209/1001348 = 439306116
  PriceCurrency      string{USD}
  Symbol             string{RIU5}
  Description         string{Фьючерсный контракт RTS-9.15}
  SecurityType       string{FUT}
  StdMaturity         string{20150914}
  FOSMarketId        RTSX
  ContractMultiplier float64{1}
  CFICode            string{FXXXXX}
  MarketSegmentID    string{F}
  InternalCreationDate Timestamp{2014-07-08 09:43:18:172}
  InternalModificationDate Timestamp{2014-07-08 09:43:18:172}
  InternalSourceId    uint16{55}
  InternalEntitlementId int32{1160}
  LocalCodeStr        string{167244614}
  PriceIncrement_static float64{10}
  MaturityYear        uint16{2015}
  MaturityMonth        uint8{9}
  MaturityDay          uint8{14}
  OperatingMIC         string{RTSX}
```

1.2.3. Indices

The sample below illustrates the details of an index:

```
instr # 209/1001000 = 439305768
  Symbol                string{USD}
  SecurityType           string{INDEX}
  FOSMarketId            RTSX
  CFICode                string{TIXXXX}
  MarketSegmentID        string{I}
  InternalCreationDate    Timestamp{2014-07-08 09:43:15:269}
  InternalModificationDate Timestamp{2014-07-08 09:43:15:269}
  InternalSourceId        uint16{55}
  InternalEntitlementId    int32{1178}
  LocalCodeStr            string{342}
  OperatingMIC            string{RTSX}
```

1.2.4. Options

The sample below illustrates the details of an option:

```
instr # 209/1024179 = 439328947
  PriceCurrency          string{USD}
  Symbol                  string{BR37BI5}
  Description              string{Сентябрьский Марж.Амер.Салл.37 Фьюч.контр BR-9.15}
  SecurityType             string{OPT}
  StrikePrice              float64{37}
  FOSMarketId              RTSX
  ContractMultiplier      float64{1}
  CFICode                  string{OCAFPX}
  MarketSegmentID          string{0}
  InternalCreationDate      Timestamp{2015-08-12 14:50:53:593}
  InternalModificationDate  Timestamp{2015-08-12 14:50:53:593}
  InternalSourceId          uint16{55}
  InternalAggregationId     uint16{55}
  InternalEntitlementId      int32{1160}
  LocalCodeStr              string{197450063}
  PriceIncrement_static    float64{0.01}
  UnderlyingLocalCodeStr    string{190978630}
  OperatingMIC              string{RTSX}
```


1.2.5. Bonds

The sample below illustrates the details of a bond:

```
instr # 372/1002289 = 781142833
  PriceCurrency      string{XXX}
  Symbol             string{INCA03}
  Description         string{ІНСАЙТ КАПІТАЛ, оді серія С, погашення 03.06.2017}
  SecurityType       string{CS}
  FOSMarketId        UKEX
  ContractMultiplier float64{1}
  CFICode            string{DBXXX}
  MarketSegmentID    string{SGTB}
  InternalCreationDate Timestamp{2015-02-01 21:02:16:801}
  InternalModificationDate Timestamp{2015-05-21 14:50:33:403}
  InternalSourceId    uint16{59}
  InternalEntitlementId int32{1158}
  LocalCodeStr       string{1410387}
  PriceIncrement_static float64{0.01}
  OperatingMIC        string{UKEX}
```

1.2.6. Multilegs

The sample below illustrates the details of a multileg:

```
instr # 209/1021818 = 439326586
  PriceCurrency      string{USD}
  Symbol             string{BRN5BRQ5}
  Description         string{календарный спред BR-7.15-8.15}
  SecurityType       string{MLEG}
  StdMaturity        string{20150716}
  FOSMarketId        RTSX
  ContractMultiplier float64{10}
  CFICode            string{FXXXXX}
  NbLegs             uint8{2}
  MarketSegmentID    string{F}
  InternalCreationDate Timestamp{2015-06-16 14:50:03:597}
  InternalModificationDate Timestamp{2015-06-16 14:50:03:597}
  InternalSourceId    uint16{55}
  InternalAggregationId uint16{55}
  InternalEntitlementId int32{1160}
  LocalCodeStr       string{194151238}
  PriceIncrement_static float64{0.01}
  MaturityYear        uint16{2015}
  MaturityMonth       uint8{7}
  MaturityDay         uint8{16}
  OperatingMIC        string{RTSX}
  LegFOSInstrumentCode_1 uint32{189987910}
  LegFOSInstrumentCode_2 uint32{188382534}
  LegRatioQty_1       float64{1}
  LegRatioQty_2       float64{1}
  LegFIXSide_1        '1'=Buy
  LegFIXSide_2        '2'=Sell
```

1.2.7. Commodities

The sample below illustrates the details of a commodity:

```
instr # 489/1001915 = 1026509243
  PriceCurrency      string{KZT}
  Symbol             string{W4CDAP}
  Description         string{Пшеница 4-го класса. Цена за тонну с НДС. Условия
поставки DAP.}
  SecurityType       string{COMMODITY}
  FOSMarketId        ETSC
  ContractMultiplier float64{1}
  CFICode            string{TTAXXX}
  MarketSegmentID    string{SAG_CLS}
  InternalCreationDate Timestamp{2015-05-12 14:50:25:452}
  InternalModificationDate Timestamp{2015-05-12 14:50:25:452}
  InternalSourceId    uint16{60}
  InternalEntitlementId int32{1159}
  LocalCodestr        string{780883}
  PriceIncrement_static float64{1}
  OperatingMIC        string{ETSC}
```

1.3. Specific Referential Tags

The following sections describe the specific referential tags available on the RTS FAST market data stream:

- [1.3.1. StdMaturity](#)
- [1.3.2. ContractMultiplier](#)
- [1.3.3. MarketSegmentID](#)
- [1.3.4. OperatingMIC](#).

1.3.1. StdMaturity

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

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

Table 2 StdMaturity – technical implementation in FeedOS

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

1.3.2. ContractMultiplier

The values of the referential tag **ContractMultiplier** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Referential* to specify the amount of underlying asset represented by each derivative contract.

FeedOS implementation of the ContractMultiplier is described in the table below:

Table 3 ContractMultiplier – technical implementation in FeedOS

Component	Value	Description
Tag Name	ContractMultiplier	FeedOS tag name.
Numeric ID	231	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	Float64	String data type.
Format / Possible values	<i>[Exchange Specific Value]</i>	An exchange specific value , specifying the amount of underlying asset represented by each derivative contract.

1.3.3. MarketSegmentID

The values of the referential tag **MarketSegmentID** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Referential* to detail the ID of the market segment.

FeedOS implementation of the tag MarketSegmentID is described below:

Table 4 MarketSegmentID – technical implementation in FeedOS

Component	Value	Description
Tag Name	MarketSegmentID	FeedOS tag name.
Numeric ID	1300	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	<i>[Exchange Specific Value]</i>	An exchange specific value , detailing the ID of the market segment.

1.3.4. OperatingMIC

The values of the referential tag **OperatingMIC** conveyed on the RTS FAST 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 the table below:

Table 5 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	<i>[Exchange Specific Value]</i>	An exchange specific value , specifying the parent MIC.
Possible Values	RTSX	Moscow Exchange – Derivatives and Classica Market

2. Quotation Data

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

```
InstrumentStatusL1
-- 209/1001348
    BID: 83150      20
    ASK: 83160      8
    LastPrice                float64{83160}
    LastTradeQty             float64{1}
    DailyHighPrice           float64{83800}
    DailyLowPrice            float64{82100}
    DailyTotalVolumeTraded   float64{506645}
    DailyTotalAssetTraded    float64{32528816590}
    LastTradePrice           float64{83160}
    LastTradeTimestamp       Timestamp{2015-08-14 12:50:50:782}
    InternalDailyOpenTimestamp Timestamp{2015-08-13 21:22:13:410}
    InternalDailyCloseTimestamp Timestamp{2015-08-13 21:22:13:410}
    InternalDailyHighTimestamp Timestamp{2015-08-14 11:30:40:775}
    InternalDailyLowTimestamp Timestamp{2015-08-14 07:00:00:406}
    InternalPriceActivityTimestamp Timestamp{2015-08-14 12:50:50:855}
    LowLimitPrice            float64{79010}
    HighLimitPrice           float64{87510}
    TradingStatus            17=ReadyToTrade
    TradingSessionId         int8{1}
    LastOffBookTradePrice    float64{82270}
    LastOffBookTradeQty      float64{12000}
    LastOffBookTradeTimestamp Timestamp{2015-08-14 12:09:46:099}
    SessionTotalVolumeTraded float64{392004}
    SessionOpeningPrice      float64{82300}
    PreviousSessionClosingPrice float64{82270}
    SessionHighPrice         float64{83800}
    SessionLowPrice          float64{82100}
    SessionVWAPPrice         float64{82880}
    SessionTotalAssetTraded  float64{32528816590}
    SessionClosingPrice      float64{83260}
    DailyOpeningPrice        float64{82300}
    PreviousDailyTotalVolumeTraded float64{90629}
    PreviousDailyTotalAssetTraded float64{7473665350}
    PreviousDailyClosingPrice float64{82270}
    PreviousBusinessDay      Timestamp{2015-08-14}
    CurrentBusinessDay       Timestamp{2015-08-13}
    PreviousDailySettlementPrice float64{82270}
    DailyTotalOffBookVolumeTraded float64{24000}
    DailyTotalOffBookAssetTraded float64{1974480000}
    OpenInterest             float64{433880}
    InternalDailyClosingPriceType char{a}
    SettlementPriceDate      Timestamp{2015-08-13}
```

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 RTS FAST 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 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	<i>[Exchange specific value]</i>	An exchange specific value , concerning 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 the specific quotation tags available on RTS FAST market data stream:

- [2.3.1. Trade Conditions](#)
- [2.3.2. Other Values.](#)

2.3.1. Trade Conditions

The following sections describe the trade conditions available on the RTS FAST market data stream:

- [2.3.1.1. TradeID.](#)

2.3.1.1. TradeID

Each time a trade occurs, the values of the quotation context tag **TradeID** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Context* to detail the unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty:

- 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 7 TradeID – technical implementation in FeedOS

Component	Value	Description
Tag Name	TradeID	FeedOS tag name.
Numeric ID	1003	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	String	String data type.
Format / Possible Values	<i>[Exchange specific value]</i>	An exchange specific value , detailing the unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.

2.3.2. Other Values

The following sections describe the other values available on the RTS FAST market data stream:

- [2.3.2.1. LowLimitPrice](#)
- [2.3.2.2. HighLimitPrice](#)
- [2.3.2.3. TradingSessionId](#)
- [2.3.2.4. SessionTotalOffBookAssetTraded](#)
- [2.3.2.5. SessionTotalOffBookVolumeTraded](#)
- [2.3.2.6. SessionTotalVolumeTraded](#)
- [2.3.2.7. PreviousSessionClosingPrice](#)
- [2.3.2.8. SessionHighPrice](#)
- [2.3.2.9. SessionLowPrice](#)
- [2.3.2.10. SessionVWAPPrice](#)
- [2.3.2.11. SessionTotalAssetTraded](#)
- [2.3.2.12. DailySettlementPrice](#)
- [2.3.2.13. CurrentBusinessDay](#)
- [2.3.2.14. OpenInterest](#)
- [2.3.2.15. InternalDailyClosingPriceType](#)
- [2.3.2.16. PriceActivityMarketTimestamp](#)
- [2.3.2.17. SettlementPriceDate.](#)

2.3.2.1. LowLimitPrice

The values of the quotation tag **LowLimitPrice** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the low limit of a 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 `LowLimitPrice` is described in the following table:

Table 8 LowLimitPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>LowLimitPrice</code>	FeedOS tag name.
Numeric ID	1148	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	<code>Float64</code>	Float64 data type.
Format / Possible Values	<i>[Exchange specific value]</i>	An exchange specific value , indicating the low limit of a price.

2.3.2.2. HighLimitPrice

The values of the quotation tag `HighLimitPrice` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the high limit of a 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 `HighLimitPrice` is described in the following table:

Table 9 HighLimitPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>HighLimitPrice</code>	FeedOS tag name.
Numeric ID	1149	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	<code>Float64</code>	Float64 data type.
Format / Possible Values	<i>[Exchange specific value]</i>	An exchange specific value , indicating the high limit of a price.

2.3.2.3. TradingSessionId

The values of the quotation tag `TradingSessionId` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the ID of the current trading session:

- 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 `TradingSessionId` is described in the following table:

Table 10 TradingSessionId – technical implementation in FeedOS

Component	Value	Description
Tag Name	TradingSessionId	FeedOS tag name.
Numeric ID	9101	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	Int8	Int8 data type.
Format / Possible Values	<i>[Exchange Specific Value]</i>	An exchange specific value , indicating the ID of the current trading session.

2.3.2.4. SessionTotalOffBookAssetTraded

The values of the quotation tag `SessionTotalOffBookAssetTraded` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the total number of assets traded off book during the current trading session:

- 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 `SessionTotalOffBookAssetTraded` is described in the table below:

Table 11 SessionTotalOffBookAssetTraded – technical implementation in FeedOS

Component	Value	Description
Tag Name	SessionTotalOffBookAssetTraded	FeedOS tag name.
Numeric ID	9114	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	<i>[Exchange Specific Value]</i>	An exchange specific value , specifying the total number of assets traded off book during the current trading session.

2.3.2.5. SessionTotalOffBookVolumeTraded

The values of the quotation tag `SessionTotalOffBookVolumeTraded` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the total volume traded off book during the current trading session:

- 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 `SessionTotalOffBookVolumeTraded` is described in the table below:

Table 12 SessionTotalOffBookVolumeTraded – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>SessionTotalOffBookVolumeTraded</code>	FeedOS tag name.
Numeric ID	9115	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the total volume traded off book during the current trading session.

2.3.2.6. SessionTotalVolumeTraded

The values of the quotation tag `SessionTotalVolumeTraded` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the total volume traded during the current trading session:

- 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 `SessionTotalVolumeTraded` is described in the table below:

Table 13 SessionTotalVolumeTraded – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>SessionTotalVolumeTraded</code>	FeedOS tag name.
Numeric ID	9120	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the total volume traded during the current trading session.

2.3.2.7. PreviousSessionClosingPrice

The values of the quotation tag `PreviousSessionClosingPrice` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the closing price value of the previous trading session:

- 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 `PreviousSessionClosingPrice` is described in the table below:

Table 14 PreviousSessionClosingPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>PreviousSessionClosingPrice</code>	FeedOS tag name.
Numeric ID	9122	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the closing price value of the previous trading session.

2.3.2.8. SessionHighPrice

The values of the quotation tag `SessionHighPrice` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the highest price value of the current trading session:

- 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 `SessionHighPrice` is described in the table below:

Table 15 SessionHighPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>SessionHighPrice</code>	FeedOS tag name.
Numeric ID	9124	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the highest price value of the current trading session.

2.3.2.9. SessionLowPrice

The values of the quotation tag `SessionLowPrice` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the lowest price value of the current trading session:

- 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 **SessionLowPrice** is described in the table below:

Table 16 SessionLowPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	SessionLowPrice	FeedOS tag name.
Numeric ID	9125	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the lowest price value of the current trading session.

2.3.2.10. SessionVWAPrice

The values of the quotation tag **SessionVWAPrice** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the volume-weighted average price value of the current trading session:

- 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 **SessionVWAPrice** is described in the table below:

Table 17 SessionVWAPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	SessionVWAPrice	FeedOS tag name.
Numeric ID	9126	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the volume-weighted average price value of the current trading session.

2.3.2.11. SessionTotalAssetTraded

The values of the quotation tag **SessionTotalAssetTraded** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the total number of assets traded during the current trading session:

- 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 `SessionTotalAssetTraded` is described in the table below:

Table 18 SessionTotalAssetTraded – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>SessionTotalAssetTraded</code>	FeedOS tag name.
Numeric ID	9127	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the total number of assets traded during the current trading session.

2.3.2.12. DailySettlementPrice

The values of the quotation tag `DailySettlementPrice` conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to specify the value of the daily settlement 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 `DailySettlementPrice` is described in the table below:

Table 19 DailySettlementPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>DailySettlementPrice</code>	FeedOS tag name.
Numeric ID	9133	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	<i>[Exchange specific value]</i>	An exchange specific value , specifying the value of the daily settlement price.

2.3.2.13. CurrentBusinessDay

Each time the trade date changes, the values of the quotation tag `CurrentBusinessDay` conveyed on the RTS FAST 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 `CurrentBusinessDay` is described in the following table:

Table 20 `CurrentBusinessDay` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>CurrentBusinessDay</code>	FeedOS tag name.
Numeric ID	9144	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	TimeStamp	TimeStamp data type.
Format / Possible Values	<i>[Exchange Specific value]</i>	<p>An exchange specific value, detailing the date and time of the trade. It is set in the evening session, at 18:45 Exchange local time (MSK), for the next business day.</p> <p>The schedule of trading on RTS is the following:</p> <ul style="list-style-type: none"> • 10.00 - 14.00 Beginning of the main trading session • 14.00 - 14.03 Intraday clearing session • 14.03 - 18.45 Ending of the main trading session • 18.45 - 19.00 Evening clearing session • 19.00 - 23.50 Evening additional trading session

2.3.2.14. OpenInterest

The values of the quotation tag **OpenInterest** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the amount of derivative contracts that have not been settled in the immediately previous time period for a specific underlying security:

- 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 `OpenInterest` is described in the table below:

Table 21 `OpenInterest` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>OpenInterest</code>	FeedOS tag name.
Numeric ID	9150	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	<i>[Exchange Specific value]</i>	<p>An exchange specific value, detailing the amount of derivative contracts that have not been settled in the immediately previous time period for a specific underlying security.</p>

2.3.2.15. InternalDailyClosingPriceType

The values of the quotation tag **InternalDailyClosingPriceType** conveyed on the RTS FAST 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 22 InternalDailyClosingPriceType – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>InternalDailyClosingPriceType</code>	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	<i>[Internal Specific value]</i>	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).

2.3.2.16. PriceActivityMarketTimestamp

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

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

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

Table 23 PriceActivityMarketTimestamp – technical implementation in FeedOS

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

2.3.2.17. SettlementPriceDate

The values of the quotation tag **SettlementPriceDate** conveyed on the RTS FAST market data stream are disseminated via FeedOS data stream in *Other Values* to indicate the date of the settlement 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 **SettlementPriceDate** is described in the table below:

Table 24 SettlementPriceDate – technical implementation in FeedOS

Component	Value	Description
Tag Name	SettlementPriceDate	FeedOS tag name.
Numeric ID	9380	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	<i>[Exchange Specific Value]</i>	An exchange specific value , indicating the date of the settlement price.

2.4. MBL and MBO Data *

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

3. Closing Price

The closing price is the last trade price upon close, as provided by the exchange. There settlement price is handled when provided by the market.

4. Special Behavior

The currency price tag of some instruments is expressed in **USR**. This means that the Price Step Value is indicated in Russian Rubles calculated using US Dollar exchange rate, which is based on MICEX USD/RUB rate while it is trading on MICEX, and based on Reuters rate when USD/RUB currency pair trade session on MICEX is closed.

Sometimes, trades outside bid/ask spread can occur. This behavior is acknowledged by the 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.

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>.