



S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

WARSAW

Reference n°: 20150812 – 28039 – 28053 – 28067

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: WARSAW
Reference 20150812 – 28039 – 28053 – 28067
August 12, 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™ WARSAW 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. Warrants	4
1.2.3. Indices	5
1.2.4. Futures	6
1.2.5. Options	7
1.3. Referential Tags	7
1.3.1. MarketSegmentID	8
1.3.2. MarketSegmentDesc	8
1.3.3. SecurityTradingId	9
1.3.4. Operating MIC and Segment MIC	9
1.3.5. MARKET_EURONEXT_InstrumentGroupCode	10
2. Quotation Data	10
2.1. Quotation Values	11
2.2. TradingStatus	11
2.3. Specific Quotation Tags	12
2.3.1. Trade Conditions	12
2.3.1.1. MARKET_EURONEXT_CrossOrderIndicator	12
2.3.1.2. MARKET_EURONEXT_TradeOffExchangeFlag	13
2.3.1.3. MARKET_EURONEXT_OpeningTradeIndicator	14
2.3.1.4. MARKET_EURONEXT_TradeTypeOfOperation	14
2.3.2. Other Values	15
2.3.2.1. MARKET_EURONEXT_HaltReason	15
2.3.2.2. MARKET_EURONEXT_ClassState	15
2.4. MBL and MBO Data	16
3. Official Closing Price	16
4. Finding the Latest Information	17



FEEDOS™ WARSAW FEED DESCRIPTION

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

The topics this feed description covers include:

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

1. Referential Data

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

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

1.1. Available Markets and Branches

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

1.1.1. Markets

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

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

FeedOS Market ID	Market
XWAR	Warsaw Stock Exchange
WDER	Warsaw Stock Exchange - Financial Derivatives

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

```
MARKETS
market # 198      CC=PL/POLAND/WARSAW,DESCR=WARSAW STOCK EXCHANGE, WEB=www.wse.com.pl
    MIC = XWAR
    TimeZone = Europe/Poland
    Country = PL
    NbMaxInstruments = 2000000
market # 365      CC=PL/POLAND/WARSAW,DESCR=WARSAW STOCK EXCHANGE/FINANCIAL DERIVATIVES,
WEB=www.pochodne.gpw.pl,OLD=BOND,SEQNUM=1
    MIC = WDER
    TimeZone = Europe/Poland
    Country = PL
    NbMaxInstruments = 2000000
```

1.1.2. Branches

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

```
BRANCHES
{ XWAR CS   ESXXXX } qty: 1964
{ XWAR CS   EXXXXX } qty: 84
{ XWAR INDEX TIXXXX } qty: 34
{ XWAR MF   EUXXXX } qty: 6
{ XWAR OPT  RAXXXX } qty: 56
{ XWAR OPT  RPXXXX } qty: 48
{ XWAR OPT  RSXXXX } qty: 14
{ XWAR WAR  RWXXXX } qty: 76
{ XWAR WAR  RXXXXX } qty: 1383
{ WDER FUT  FXXXXX } qty: 810
{ WDER MF   EUXXXX } qty: 1
{ WDER OPT  OXXXXX } qty: 2496
```

1.2. Types of Instruments

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

- [1.2.1. Equities](#)
- [1.2.2. Warrants](#)
- [1.2.3. Indices](#)
- [1.2.4. Futures](#)
- [1.2.5. Options.](#)

1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 198/4391 = 415240487
  PriceCurrency      string{PLN}
  Symbol             string{BIP}
  Description         string{BIOPLANET}
  SecurityType       string{CS}
  FOSMarketId        XWAR
  CFICode            string{ESXXX}
  CountryOfIssue     string{POL}
  RoundLot           float64{1}
  SecuritySubType    string{Equity Ordinary Shares}
  StrikeCurrency     string{PLN}
  SecurityGroup       string{B5}
  MarketSegmentID    string{OT}
  MarketSegmentDesc  string{CashBlockTrading}
  InternalCreationDate Timestamp{2015-03-31 16:00:57:151}
  InternalModificationDate Timestamp{2015-08-10 03:04:10:798}
  InternalSourceId    uint16{39}
  LocalCodeStr       string{21593}
  ISIN               string{PLBIOPL00013}
  PriceIncrement_dynamic_TableId uint32{2556005}
  SecurityTradingId   string{PLBIOPL0001B}
  MBLLayersDesc       string{0}
  OperatingMIC        string{XWAR}
  SegmentMIC          string{XNCO}
  MARKET_EURONEXT_InstrumentGroupCode string{B5}
  MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice string{1}
  MARKET_EURONEXT_NominalMarketValueOfTheSecurity float64{1}
```

1.2.2. Warrants

The sample below illustrates the details of a warrant:

```
instr # 198/4658 = 415240754
  PriceCurrency      string{PLN}
  Description        string{INTSW2010082}
  SecurityType       string{WAR}
  StrikePrice        float64{2293.76}
  FOSMarketId        XWAR
  CFICode            string{RXXXXX}
  CountryOfIssue     string{POL}
  RoundLot           float64{1}
  SecuritySubType    string{Structured Product Certificates with Knock-Out}
  StrikeCurrency     string{PLN}
  SecurityGroup      string{94}
  MarketSegmentID   string{LE}
  MarketSegmentDesc  string{Leverage}
  InternalCreationDate Timestamp{2015-07-29 16:01:00:545}
  InternalModificationDate Timestamp{2015-08-12 04:40:00:728}
  InternalSourceId   uint16{39}
  InternalAggregationId uint16{39}
  LocalCodeStr       string{22826}
  ISIN               string{PLINGNV10082}
  PriceIncrement_dynamic_TableId uint32{2556004}
  SecurityTradingId  string{PLINGNV10082}
  MBLLayersDesc      string{0}
  OperatingMIC        string{XWAR}
  SegmentMIC          string{WETP}
  MARKET_EURONEXT_InstrumentGroupCode string{94}
  MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice string{1}
  MARKET_EURONEXT_NominalMarketValueOfTheSecurity float64{0}
```

1.2.3. Indices

The sample below illustrates the details of an index:

```
instr # 198/3647 = 415239743
  PriceCurrency           string{PLN}
  Description             string{WIG250}
  SecurityType           string{INDEX}
  FOSMarketId            XWAR
  CFICode                string{TIXXXX}
  CountryOfIssue         string{POL}
  SecuritySubType        string{Index}
  SecurityGroup          string{X1}
  InternalCreationDate    Timestamp{2014-03-21 17:54:57:995}
  InternalModificationDate Timestamp{2015-01-08 09:11:57:896}
  InternalHideFromLookup bool{True}
  InternalSourceId       uint16{39}
  InternalAggregationId  uint16{39}
  InternalEntitlementId   int32{1101}
  DelayedFeedMin         uint16{15}
  LocalCodeStr           string{18587}
  ISIN                   string{PL9999999326}
  PriceIncrement_dynamic_TableId uint32{2556009}
  MBLLayersDesc          string{0}
  OperatingMIC           string{XWAR}
  MARKET_EURONEXT_InstrumentGroupCode string{X1}
  MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice string{}
  MARKET_EURONEXT_NominalMarketValueOfTheSecurity float64{1143.64}
  IndexComponentFOSInstrumentCode uint32{415238774}
  IndexComponentFOSInstrumentCode_1 uint32{415239181}
  IndexComponentFOSInstrumentCode_2 uint32{415239399}
  IndexComponentFOSInstrumentCode_3 uint32{415239721}
  [...]
```


1.2.4. Futures

The sample below illustrates the details of a future:

```
instr # 365/3476 = 765463956
  PriceCurrency      string{PLN}
  Symbol             string{FUSDH16}
  Description         string{FUSDH16}
  SecurityType       string{FUT}
  FOSMarketId        WDER
  CFICode            string{FXXXXX}
  RoundLot           float64{1}
  SecuritySubType    string{Futures Futures}
  SecurityGroup      string{F3}
  MarketSegmentID    string{DN}
  MarketSegmentDesc  string{DerivativesNormalMarket}
  InternalCreationDate Timestamp{2015-03-20 17:16:02:218}
  InternalModificationDate Timestamp{2015-08-12 03:04:20:798}
  InternalSourceId    uint16{39}
  LocalCodeStr       string{21570}
  ISIN               string{PL0GF0008409}
  MaturityYear        uint16{2016}
  MaturityMonth       uint8{3}
  MaturityDay         uint8{18}
  PriceIncrement_dynamic_TableId uint32{2556004}
  SecurityTradingId   string{PL0GF0008409}
  MBLLayersDesc       string{0}
  OperatingMIC        string{XWAR}
  SegmentMIC          string{WDER}
  MARKET_EURONEXT_InstrumentGroupCode string{F3}
  MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice string{1}
  MARKET_EURONEXT_NominalMarketValueOfTheSecurity float64{0}
```

1.2.5. Options

The sample below illustrates the details of an option:

```
instr # 365/4306 = 765464786
  PriceCurrency      string{PLN}
  Symbol             string{OW20J151800}
  Description         string{OW20J151800}
  SecurityType       string{OPT}
  StrikePrice        float64{1800}
  FOSMarketId        WDER
  CFICode            string{OXXXXX}
  RoundLot           float64{1}
  SecuritySubType    string{Options Vanilla Options}
  StrikeCurrency     string{PLN}
  SecurityGroup      string{P1}
  MarketSegmentID    string{DB}
  MarketSegmentDesc  string{DerivativesBlockTrading}
  InternalCreateDate  Timestamp{2015-07-22 16:01:02:864}
  InternalModificationDate Timestamp{2015-08-10 04:40:04:427}
  InternalSourceId    uint16{39}
  InternalAggregationId uint16{39}
  LocalCodeStr       string{22778}
  ISIN               string{PL0G00065906}
  MaturityYear        uint16{2015}
  MaturityMonth       uint8{10}
  MaturityDay         uint8{16}
  PriceIncrement_dynamic_TableId uint32{2556005}
  SecurityTradingId   string{PL0G0006590B}
  MBLayersDesc        string{0}
  OperatingMIC        string{XWAR}
  SegmentMIC          string{WDER}
  MARKET_EURONEXT_InstrumentGroupCode string{P1}
  MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice string{1}
  MARKET_EURONEXT_NominalMarketValueOfTheSecurity float64{0}
```

1.3. Referential Tags

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

- [1.3.1. MarketSegmentID](#)
- [1.3.2. MarketSegmentDesc](#)
- [1.3.3. SecurityTradingId](#)
- [1.3.4. Operating MIC and Segment MIC](#)
- [1.3.5. MARKET_EURONEXT_InstrumentGroupCode.](#)

1.3.1. MarketSegmentID

The values of the referential tag **MarketSegmentID** conveyed on the WARSAW 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 2 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	<i>[Exchange specific value]</i>	An exchange specific value , detailing the ID of the market segment.
Possible Values	BA	BondSpot ATS – For BondSpot only
	BR	Retail Market – For BondSpot only
	BT	BondSpot TBS Poland – For BondSpot only
	BW	Wholesale Market – For BondSpot only
	DB	Derivatives Block Trading
	DN	Derivatives Normal Market
	IN	Investment
	IP	IPO
	IX	WSE Indices
	LE	Leverage
	NM	Cash Normal Market
	OT	Cash Block Trading
	TO	Tender Offer

1.3.2. MarketSegmentDesc

The values of the referential tag **MarketSegmentDesc** conveyed on the WARSAW market data stream are disseminated via FeedOS data stream in *Referential* to describe the market segment.

FeedOS implementation of the tag MarketSegmentDesc is described below:

Table 3 MarketSegmentDesc – technical implementation in FeedOS

Component	Value	Description
Tag Name	MarketSegmentDesc	FeedOS tag name.
Numeric ID	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 data type.
Format	<i>[Exchange specific value]</i>	An exchange specific value , describing the market segment.

Table 3 MarketSegmentDesc – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	BondSpotATS	For BondSpot only.
	BondSpotTBSPoland	
	RetailMarket	
	WholesaleMarket	
	CashBlockTrading	For all the other instruments.
	CashNormalMarket	
	DerivativesBlockTrading	
	DerivativesNormalMarket	
	Investment	
	IPO	
	Leverage	
	TenderOffer	
	WSEIndices	

1.3.3. SecurityTradingId

The values of the referential tag **SecurityTradingId** conveyed on the WARSAW market data stream are disseminated via FeedOS data stream in *Referential* to specify the instrument Security Code.

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

Table 4 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	<i>[Exchange specific value]</i>	An exchange specific value , detailing the instrument Security Code.

1.3.4. Operating MIC and Segment MIC

The values of the referential tags **Operating MIC** and **Segment MIC** conveyed on the WARSAW market data stream are disseminated via FeedOS data stream in *Referential* to specify the parent and child MIC.

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

Table 5 `OperatingMIC` and `SegmentMIC` – technical implementation in FeedOS

Component	Value		Description
Tag Name	<code>OperatingMIC</code>	<code>SegmentMIC</code>	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	<i>[Exchange Specific value]</i>	<i>[Exchange Specific value]</i>	An exchange specific value , specifying the parent and child MICs.
Possible Values	XWAR	TBSP	Bonds/ Bondspot Treasury Bond Market
	XWAR	WDER	Financial Derivatives
	XWAR	WETP	ETPs
	XWAR	WIND	Indices
	XWAR	XNCO	Equities / New Connect - MTF

1.3.5. MARKET_EURONEXT_InstrumentGroupCode

The referential tag `MARKET_EURONEXT_InstrumentGroupCode` is disseminated via FeedOS market data stream in *Referential*, when new details about the Instrument Group are available.

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

Table 6 `MARKET_EURONEXT_InstrumentGroupCode` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>MARKET_EURONEXT_InstrumentGroupCode</code>	FeedOS tag name.
Numeric ID	11050	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 characteristics of the Instrument Group.

2. Quotation Data

The sections below describe the characteristics of the quotation data on the WARSAW 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 WARSAW market data stream:

```
InstrumentStatusL1
-- 198/2485
    BID: 83.51      1570    @3
    ASK: 83.53      178     @2
    LastPrice                float64{83.53}
    LastTradeQty              float64{51}
    DailyHighPrice            float64{84.72}
    DailyLowPrice             float64{83.06}
    DailyTotalVolumeTraded    float64{371966}
    DailyTotalAssetTraded     float64{31245454.53}
    LastTradePrice            float64{83.53}
    LastTradeTimestamp         Timestamp{2015-08-12 09:07:00:185}
    InternalDailyOpenTimestamp Timestamp{2015-08-12 07:00:00:068}
    InternalDailyCloseTimestamp Timestamp{2015-08-11 15:05:00:021}
    InternalDailyHighTimestamp Timestamp{2015-08-12 07:25:18:378}
    InternalDailyLowTimestamp  Timestamp{2015-08-12 08:53:05:122}
    InternalPriceActivityTimestamp Timestamp{2015-08-12 09:07:00:324}
    TradingStatus              17=ReadyToTrade
    DailyOpeningPrice          float64{84.7}
    PreviousDailyTotalVolumeTraded float64{1215739}
    PreviousDailyTotalAssetTraded float64{102895034.04}
    PreviousDailyClosingPrice   float64{84.77}
    PreviousBusinessDay         Timestamp{2015-08-11}
    CurrentBusinessDay          Timestamp{2015-08-12}
    LastAuctionPrice            float64{84.7}
    LastAuctionVolume           float64{1209}
    InternalLastAuctionTimestamp Timestamp{2015-08-12 06:59:55:923}
    PriceActivityMarketTimestamp Timestamp{2015-08-12 09:07:00:308}
    MARKET_EURONEXT_ClassState  string{COCO}
```

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** in the WARSAW 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 table below:

Table 7 Trading Status of the WARSAW market data stream – technical implementation in FeedOS

Component	Value	Description
Tag Name	TradingStatus	FeedOS tag name.
Numeric ID	9100	FeedOS unique ID broadcast on the S&P Capital IQ Real-Time Solutions data stream. It is the numeric equivalent of the tag name.
Type	Enum	Enumeration data type.
Format	<i>[Exchange specific value]</i>	An exchange specific value , as described below, 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 section describe the specific quotation tags available on the WARSAW market data stream:

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

2.3.1. Trade Conditions

The sections below describe the trade conditions on the WARSAW market data stream:

- [2.3.1.1. MARKET_EURONEXT_CrossOrderIndicator](#)
- [2.3.1.2. MARKET_EURONEXT_TradeOffExchangeFlag](#)
- [2.3.1.3. MARKET_EURONEXT_OpeningTradeIndicator](#)
- [2.3.1.4. MARKET_EURONEXT_TradeTypeOfOperation.](#)

2.3.1.1. MARKET_EURONEXT_CrossOrderIndicator

Each time a cross order occurs, the values of the quotation tag **MARKET_EURONEXT_CrossOrderIndicator** conveyed on the WARSAW 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_EURONEXT_CrossOrderIndicator` is described in the table below:

Table 8 `MARKET_EURONEXT_CrossOrderIndicator` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>MARKET_EURONEXT_CrossOrderIndicator</code>	FeedOS tag name.
Numeric ID	15050	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	<i>[Exchange Specific Value]</i>	An exchange specific value , indicating the presence of a cross order.
Possible Values	0	The trade does not stem from a Cross Order.
	1	The trade stems from a Cross Order.
	4	Valuation trade.
	5	Internalized Trade
	6	Internalized Crossed Trade

2.3.1.2. `MARKET_EURONEXT_TradeOffExchangeFlag`

Each time a trade relates to a block or a negotiated deal following MiFID rules, the values of the quotation tag `MARKET_EURONEXT_TradeOffExchangeFlag` conveyed on the WARSAW 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_EURONEXT_TradeOffExchangeFlag` is described in the table below:

Table 9 `MARKET_EURONEXT_TradeOffExchangeFlag` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>MARKET_EURONEXT_TradeOffExchangeFlag</code>	FeedOS tag name.
Numeric ID	15053	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	<i>[Exchange Specific Value]</i>	An exchange specific value , specifying whether the trade relates to a block or a deal negotiated on MiFID rules.
Possible Values	B	Orders from the Book
	O	Block trade
	T	Tender Offer trade
	S	BISO trade
	R	Trade reporting (for future use)

2.3.1.3. MARKET_EURONEXT_OpeningTradeIndicator

Each time a trade occurs, the values of the quotation tag **MARKET_EURONEXT_OpeningTradeIndicator** conveyed on the WARSAW market data stream are disseminated via FeedOS data stream in *Context* to specify the moment when the trade took place:

- 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_EURONEXT_OpeningTradeIndicator** is described in the table below:

Table 10 MARKET_EURONEXT_OpeningTradeIndicator – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_OpeningTradeIndicator	FeedOS tag name.
Numeric ID	15055	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	<i>[Exchange Specific Value]</i>	An exchange specific value , specifying the moment when the trade took place.
Possible Values	O	Opening Note: The first trade of the day will always have a value of 'O' – even if it occurs during the core session.
	S	Core Session

2.3.1.4. MARKET_EURONEXT_TradeTypeOfOperation

Each time a trade occurs, the values of the quotation tag **MARKET_EURONEXT_TradeTypeOfOperation** conveyed on the WARSAW 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_EURONEXT_TradeTypeOfOperation** is described in the table below:

Table 11 MARKET_EURONEXT_TradeTypeOfOperation – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_TradeTypeOfOperation	FeedOS tag name.
Numeric ID	15056	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	<i>[Exchange Specific Value]</i>	An exchange specific value , detailing the type of operation.
Possible Values	B	Orders from the book
	I	IPO trade

2.3.2. Other Values

The following sections describe the specific quotation tags available on the WARSAW market data stream:

- [2.3.2.1. MARKET_EURONEXT_HaltReason](#)
- [2.3.2.2. MARKET_EURONEXT_ClassState](#).

2.3.2.1. MARKET_EURONEXT_HaltReason

Each time an instrument is halted from trading, the values of the quotation tag **MARKET_EURONEXT_HaltReason** conveyed on the WARSAW 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_EURONEXT_HaltReason** is described in the table below:

Table 12 MARKET_EURONEXT_HaltReason – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_HaltReason	FeedOS tag name.
Numeric ID	14550	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	<i>[Exchange specific value]</i>	An exchange specific value , detailing the reason of halting for an instrument.
Possible Values	Null or Space	Instrument not halted or information not available.
	A	Number of Orders Alert
	C	Trade price outside dynamic collars
	G	Regulatory halting by market operations.
	M	Manually halting by market operations
	R	Halted. No liquidity provider
	S	Trade price outside static collars
	W	Regulatory halting, deactivating barrier reached
	X	Regulatory halting, underlying halted

2.3.2.2. MARKET_EURONEXT_ClassState

Each time the state of a group of instruments changes, the values of the quotation tag **MARKET_EURONEXT_ClassState** conveyed on the WARSAW 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_EURONEXT_ClassState` is described in the table below:

Table 13 `MARKET_EURONEXT_ClassState` – technical implementation in FeedOS

Component	Value	Description
Tag Name	<code>MARKET_EURONEXT_ClassState</code>	FeedOS tag name.
Numeric ID	14551	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	<i>[Exchange specific value]</i>	An exchange specific value , indicating the state of a group of instruments.
Possible Values	EAMO	Early Monitoring
	COCA	Core Call
	COAU	Core Auction
	COCO	Core Continuous
	CLCA	Closing Call
	CLAU	Closing Auction
	TAL	Trading At Last
	COMO	Core Monitoring
	LAMO	Late Monitoring
	CLSD	Closed
	HALT	Halted

2.4. MBL and MBO Data *

The MBL book has a 5-level depth. There is no MBO.

3. Official Closing Price

The closing price is the last trade price upon close, as provided by the exchange. If the instrument has an auction phase, the market sends the last auction price, which becomes the closing price. When a stock splits, the closing price is adjusted after the closing. The settlement price is handled when provided 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.

4. 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>.