S&P CAPITAL IQ
McGRAW HILL FINANCIAL

S&P Capital IQ Real-Time Solutions

FeedOS™ Feed Description

EURONEXT UTP

Reference n°: 20150701 – 26648 – 27200 – 27204

S&P Capital IQ Real-Time Solutions
FeedOS™ Feed Description: EURONEXT UTP
Reference 20150701 – 26648 – 27200 – 27204
July 24, 2015

France
52 Rue de la Victoire
75009 Paris
France
Tel: +33 (0) 1 73 02 32 11

United States
55 Water Street, 44th floor
New York, NY 10041
United States of America
Tel: +1-(212)-438-4346

130 East Randolph
One Prudential Plaza, Suite 2900
Chicago, IL 60601
United States of America
Tel: +1-(312)-233-7129

United Kingdom
20 Canada Square
Canary Wharf
London E14 5LH
United Kingdom
Tel: +44 (0) 203 107 1676

Singapore
12 Marina Boulevard
#23-01 Marina Bay
Financial Centre Tower 3
Singapore 018982
Tel: +65 6530 6546

www.spcapitaliq.com

TABLE OF CONTENTS

FEEDOS™ EURONEXT UTP 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 EURONEXT UTP 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 EURONEXT UTP market data stream, in terms of:

1.1. Available Markets and Branches

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

1.1.1. Markets

The EURONEXT UTP market data stream broadcasts informations about the following markets:

Table 1 List of markets available on the EURONEXT UTP market data stream

FeedOS Market ID	Market
XBRU	Euronext Brussels
XMLI	Euronext Paris – Marché Libre
XPAR	Euronext Paris
XKUX	Luxembourg Stock Exchange
XLIS	Euronext Lisboa
XAMS	Euronext Amsterdam
ENXB	NYSE Euronext - EasyNext
WQXL	NYSE Euronext - Market without Quotations Lisbon
ENXL	NYSE Euronext - EasyNext Lisbon
TNLB	NYSE Euronext - Trading Facility
MLXB	NYSE Euronext - Marché Libre Brussels
ALXP	NYSE Euronext - AlterNext Paris
ALXB	NYSE Euronext - AlterNext Brussels
ALXA	NYSE Euronext - Alternext Amsterdam
TNLA	NYSE Euronext - Traded but not listed Amsterdam
XHFT	NYSE Arca Europe
misc	S&P Capital IQ Real-Time Solutions internal MIC for TCS Trade Report Platform. Available for instruments traded on other markets and reported on Euronext.

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

```
MARKETS
market # 32     CC=BE/BELGIUM/BRUSSELS,DESCR=EURONEXT BRUSSELS,WEB=www.euronext.com
    MIC = XBRU
    TimeZone = Europe/Brussels
    Country = BE
    NbMaxInstruments = 2000000
market # 79     CC=FR/FRANCE/PARIS,DESCR=EURONEXT PARIS S.A. - MARCHE LIBRE,
WEB=www.euronext.com
    MIC = XMLI
    TimeZone = Europe/Paris
    Country = FR
    NbMaxInstruments = 2000000
market # 81     CC=FR/FRANCE/PARIS,DESCR=EURONEXT PARIS S.A., WEB=www.euronext.com
    MIC = XPAR
    TimeZone = Europe/Paris
    Country = FR
    NbMaxInstruments = 2000000
market # 163    CC=LU/LUXEMBOURG/LUXEMBOURG,DESCR=LUXEMBOURG STOCK EXCHANGE,
WEB=www.bourse.lu
    MIC = XLUX
    TimeZone = Europe/Luxembourg
    Country = LU
    NbMaxInstruments = 2000000                                    (see next page)
```

```
MARKETS CONTINUED
market # 202    CC=PT/PORTUGAL/LISBOA,DESCR=EURONEXT LISBOA, WEB=www.euronext.pt
    MIC = XLIS
    TimeZone = Europe/Lisboa
    Country = PT
    NbMaxInstruments = 2000000
market # 265    CC=NL/THE NETHERLANDS/AMSTERDAM,DESCR=EURONEXT AMSTERDAM, WEB=www.aex.nl
    MIC = XAMS
    TimeZone = Europe/Amsterdam
    Country = NL
    NbMaxInstruments = 2000000
market # 328    CC=BE/BELGIUM/BRUSSELS,DESCR=NYSE EURONEXT - EASY NEXT, WEB=www.euronext.com
    MIC = ENXB
    TimeZone = Europe/Brussels
    Country = BE
    NbMaxInstruments = 2000000
market # 354   CC=PT/PORTUGAL/LISBOA,DESCR=NYSE EURONEXT - MARKET WITHOUT QUOTATIONS LISBON,
WEB=www.euronext.com
    MIC = WQXL
    TimeZone = Europe/Lisboa
    Country = PT
    NbMaxInstruments = 2000000
market # 355    CC=PT/PORTUGAL/LISBOA,DESCR=NYSE EURONEXT - EASYNEXT LISBON,
WEB=www.euronext.com
    MIC = ENXL
    TimeZone = Europe/Lisboa
    Country = PT
    NbMaxInstruments = 2000000
market # 416    CC=BE/BELGIUM/BRUSSELS,DESCR=NYSE EURONEXT - TRADING FACILITY BRUSSELS,
WEB=www.euronext.com
    MIC = TNLB
    TimeZone = Europe/Brussels
    Country = BE
    NbMaxInstruments = 2000000
market # 429    CC=BE/BELGIUM/BRUSSELS,DESCR=NYSE EURONEXT - MARCHE LIBRE BRUSSELS,
WEB=www.euronext.com
    MIC = MLXB
    TimeZone = Europe/Brussels
    Country = BE
    NbMaxInstruments = 2000000
market # 434    CC=FR/FRANCE/PARIS,DESCR=NYSE EURONEXT - ALTERNEXT PARIS,
WEB=www.euronext.com
    MIC = ALXP
    TimeZone = Europe/Paris
    Country = FR
    NbMaxInstruments = 2000000
market # 435    CC=BE/BELGIUM/BRUSSELS,DESCR=NYSE EURONEXT - ALTERNEXT BRUSSELS,
WEB=www.euronext.com
    MIC = ALXB
    TimeZone = Europe/Brussels
    Country = BE
    NbMaxInstruments = 2000000                                          (see next page)
```

```
MARKETS CONTINUED
market # 436    CC=NL/THE NETHERLANDS/AMSTERDAM,DESCR=NYSE EURONEXT - ALTERNEXT AMSTERDAM,
WEB=www.euronext.com
    MIC = ALXA
    TimeZone = Europe/Amsterdam
    Country = NL
    NbMaxInstruments = 2000000
market # 438    CC=NL/THE NETHERLANDS/AMSTERDAM,DESCR=NYSE EURONEXT - TRADED BUT NOT LISTED
AMSTERDAM, WEB=www.euronext.com
    MIC = TNLA
    TimeZone = Europe/Amsterdam
    Country = NL
    NbMaxInstruments = 2000000
market # 446    CC=NL/THE NETHERLANDS/AMSTERDAM,DESCR=NYSE ARCA EUROPE, WEB=www.euronext.com
    MIC = XHFT
    TimeZone = Europe/Amsterdam
    Country = NL
    NbMaxInstruments = 2000000
market # 501    CC=unknown/unknown/unknown,DESCR=Miscellaneous, WEB=
    MIC = misc
    TimeZone = Europe/Paris
    Country = unknown
    NbMaxInstruments = 2000000
```

1.1.2. Branches

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

```
BRANCHES
    { XBRU GO   DBFSFR } qty: 5
    { XBRU GO   DBFUFN } qty: 2
    { XBRU GO   DBVUFN } qty: 11
    { XBRU GO   DBXXXX } qty: 500
    { XBRU GO   DTFXFB } qty: 2
    { XBRU GO   DYFSFB } qty: 1
    { XBRU GO   DYZTXN } qty: 51
    { XBRU INDEX MRIXXX } qty: 66
    { XBRU MTN  DTXXXX } qty: 124
    { XBRU NONE DBXXXX } qty: 315
    { XBRU NONE DCXXXX } qty: 1
    { XBRU NONE DXXTXX } qty: 1
    { XBRU NONE ESXXXX } qty: 432
    { XBRU NONE EUOGSN } qty: 1
    { XBRU NONE EUXXXX } qty: 101
    { XBRU NONE MRIXXX } qty: 8
    { XBRU NONE RAXXXX } qty: 10
    { XBRU NONE RMXXXX } qty: 174
    { XBRU NONE RSXXXX } qty: 13
    { XBRU PS   EPXXXX } qty: 2
    { XBRU PS   ERXXXX } qty: 2
    { XBRU WAR  RWSXXX } qty: 15
    { XBRU WAR  RWXXXX } qty: 3713                                          (see next page)
```

```
BRANCHES CONTINUED
    { XMLI GO   DBXXXX } qty: 15
    { XMLI MTN  DTXXXX } qty: 34
    { XMLI NONE DCXXXX } qty: 3
    { XMLI NONE ESXXXX } qty: 320
    { XMLI NONE RSXXXX } qty: 4
    { XMLI PS   ERXXXX } qty: 2
    { XMLI WAR  RWSXXX } qty: 10
    { XPAR CB   DCXXXX } qty: 1
    { XPAR GO   DBFSFB } qty: 3
    { XPAR GO   DBFUCB } qty: 1
    { XPAR GO   DBFUFB } qty: 3
    { XPAR GO   DBFUFR } qty: 1
    { XPAR GO   DBFXAB } qty: 1
    { XPAR GO   DBFXDB } qty: 1
    { XPAR GO   DBFXFB } qty: 8
    { XPAR GO   DBFXFR } qty: 1
    { XPAR GO   DBFXXB } qty: 13
    { XPAR GO   DBVGXB } qty: 2
    { XPAR GO   DBVUQB } qty: 1
    { XPAR GO   DBVUXB } qty: 1
    { XPAR GO   DBVXAB } qty: 1
    { XPAR GO   DBVXFB } qty: 7
    { XPAR GO   DBVXGB } qty: 1
    { XPAR GO   DBVXXB } qty: 9
    { XPAR GO   DBXXAX } qty: 6
    { XPAR GO   DBXXXX } qty: 838
    { XPAR GO   DTFSFB } qty: 2
    { XPAR GO   DTFXAB } qty: 1
    { XPAR GO   DTFXDB } qty: 2
    { XPAR GO   DTFXFB } qty: 16
    { XPAR GO   DTFXFR } qty: 3
    { XPAR GO   DTFXGB } qty: 1
    { XPAR GO   DTFXXB } qty: 20
    { XPAR GO   DTFXXN } qty: 1
    { XPAR GO   DTVXFB } qty: 3
    { XPAR GO   DTVXFR } qty: 2
    { XPAR GO   DYFTXX } qty: 91
    { XPAR GO   DYFUXB } qty: 15
    { XPAR GO   DYFXXB } qty: 174
    { XPAR GO   DYFXXX } qty: 784
    { XPAR GO   DYVTXX } qty: 190
    { XPAR GO   DYVXXB } qty: 510
    { XPAR GO   DYVXXX } qty: 2311
    { XPAR GO   DYXXXX } qty: 1
    { XPAR GO   DYZTXX } qty: 451
    { XPAR GO   DYZXXB } qty: 442
    { XPAR GO   DYZXXX } qty: 1034
    { XPAR GO   ESXUFB } qty: 1
    { XPAR INDEX MRIXXX } qty: 776
    { XPAR MTN  DTVXXX } qty: 19
    { XPAR MTN  DTXXXX } qty: 1996
    { XPAR NONE DBXTAX } qty: 221
    { XPAR NONE DBXTFX } qty: 42
    { XPAR NONE DBXTXX } qty: 64
    { XPAR NONE DBXXXX } qty: 216                                    (see next page)
```

```
BRANCHES CONTINUED
    { XPAR NONE DBZGFB } qty: 12
    { XPAR NONE DCXXXX } qty: 87
    { XPAR NONE DMXXFB } qty: 1
    { XPAR NONE DMXXXX } qty: 17
    { XPAR NONE DXXTXX } qty: 14
    { XPAR NONE DXXXXX } qty: 70
    { XPAR NONE ESNUFR } qty: 12
    { XPAR NONE ESXXXX } qty: 1461
    { XPAR NONE EUOGBE } qty: 2
    { XPAR NONE EUOGEE } qty: 1
    { XPAR NONE EUOGMN } qty: 1
    { XPAR NONE EUOGSB } qty: 21
    { XPAR NONE EUOGSN } qty: 52
    { XPAR NONE EUOGSR } qty: 13
    { XPAR NONE EUOGXB } qty: 106
    { XPAR NONE EUOGXN } qty: 32
    { XPAR NONE EUOGXZ } qty: 1
    { XPAR NONE EUOISB } qty: 1
    { XPAR NONE EUOISN } qty: 1
    { XPAR NONE EUOISR } qty: 4
    { XPAR NONE EUOIXB } qty: 23
    { XPAR NONE EUOIXN } qty: 2
    { XPAR NONE EUOIXR } qty: 2
    { XPAR NONE EUOMSB } qty: 1
    { XPAR NONE EUOMSN } qty: 14
    { XPAR NONE EUOMXB } qty: 244
    { XPAR NONE EUOMXN } qty: 28
    { XPAR NONE EUOXSN } qty: 3
    { XPAR NONE EUOXXN } qty: 8
    { XPAR NONE EUOXXX } qty: 13
    { XPAR NONE EUXXXX } qty: 180
    { XPAR NONE MRCXXX } qty: 45
    { XPAR NONE MRIXXX } qty: 322
    { XPAR NONE RAXXXX } qty: 1
    { XPAR NONE RMXXXX } qty: 9810
    { XPAR NONE RSXXXX } qty: 71
    { XPAR NONE RWXXXX } qty: 26
    { XPAR PS   EPXXXX } qty: 1
    { XPAR PS   ERXXXX } qty: 3
    { XPAR WAR  RWSXXX } qty: 64633
    { XPAR WAR  RWXXXX } qty: 215482
    { XLUX GO   DBXXXX } qty: 36225
    { XLUX INDEX MRIXXX } qty: 38
    { XLUX NONE DWXXXX } qty: 1
    { XLUX NONE ESXXXX } qty: 80
    { XLUX NONE EUXXXX } qty: 8458
    { XLUX NONE MRIXXX } qty: 22
    { XLUX WAR  RWMXXX } qty: 7038
    { XLIS GO   DBFUFB } qty: 6
    { XLIS GO   DBVSDR } qty: 3
    { XLIS GO   DBVUDB } qty: 1
    { XLIS GO   DBVUFB } qty: 9
    { XLIS GO   DBVUGB } qty: 1
    { XLIS GO   DBXXXX } qty: 230
    { XLIS GO   DTVUFR } qty: 1                                          (see next page)
```

```
BRANCHES CONTINUED
    { XLIS GO   DYFTXR } qty: 28
    { XLIS GO   DYFUXR } qty: 423
    { XLIS GO   DYVUXR } qty: 65
    { XLIS GO   DYXXXX } qty: 1
    { XLIS INDEX MRIXXX } qty: 39
    { XLIS MTN  DTXXXX } qty: 105
    { XLIS NONE DBXXXX } qty: 37
    { XLIS NONE DCXXXX } qty: 1
    { XLIS NONE DMXXXX } qty: 2
    { XLIS NONE ESXXXX } qty: 66
    { XLIS NONE EUOGSR } qty: 9
    { XLIS NONE EUXXXX } qty: 12
    { XLIS NONE RAXXXX } qty: 13
    { XLIS NONE RSXXXX } qty: 7
    { XLIS PS   ERXXXX } qty: 2
    { XAMS GO   DBFGQB } qty: 1
    { XAMS GO   DBFSBB } qty: 3
    { XAMS GO   DBFSFN } qty: 16
    { XAMS GO   DBFSGB } qty: 2
    { XAMS GO   DBFUQB } qty: 1
    { XAMS GO   DBFXFB } qty: 2
    { XAMS GO   DBFXXB } qty: 4
    { XAMS GO   DBFXXN } qty: 5
    { XAMS GO   DBVSBB } qty: 4
    { XAMS GO   DBVSCN } qty: 1
    { XAMS GO   DBVSGB } qty: 4
    { XAMS GO   DBVSGR } qty: 2
    { XAMS GO   DBVUQB } qty: 2
    { XAMS GO   DBVXBB } qty: 7
    { XAMS GO   DBVXFB } qty: 115
    { XAMS GO   DBVXFR } qty: 8
    { XAMS GO   DBVXGB } qty: 27
    { XAMS GO   DBVXXB } qty: 1
    { XAMS GO   DBVXXN } qty: 11
    { XAMS GO   DBXXAX } qty: 93
    { XAMS GO   DBXXFX } qty: 18
    { XAMS GO   DBXXXX } qty: 1130
    { XAMS GO   DBZSBB } qty: 4
    { XAMS GO   DTFXFB } qty: 2
    { XAMS GO   DTVXFB } qty: 1
    { XAMS GO   DYXXXB } qty: 11
    { XAMS INDEX MRIXXX } qty: 238
    { XAMS MTN  DTVXXX } qty: 23
    { XAMS MTN  DTXXXX } qty: 1278
    { XAMS NONE DBXXXX } qty: 58
    { XAMS NONE DCXXXX } qty: 3
    { XAMS NONE DXXTXX } qty: 2
    { XAMS NONE DXXXXX } qty: 28
    { XAMS NONE ESVUFN } qty: 4
    { XAMS NONE ESXXXX } qty: 290
    { XAMS NONE EUOGCR } qty: 1
    { XAMS NONE EUOGSE } qty: 6
    { XAMS NONE EUOGSR } qty: 5
    { XAMS NONE EUOGXR } qty: 16
    { XAMS NONE EUOISN } qty: 1                                    (see next page)
```

```
BRANCHES CONTINUED
    { XAMS NONE EUOISR } qty: 19
    { XAMS NONE EUOIXR } qty: 9
    { XAMS NONE EUOMSN } qty: 2
    { XAMS NONE EUOMXB } qty: 1
    { XAMS NONE EUOXXR } qty: 3
    { XAMS NONE EUOXXX } qty: 5
    { XAMS NONE EUXXXX } qty: 415
    { XAMS NONE MRIXXX } qty: 55
    { XAMS NONE RAXXXX } qty: 1
    { XAMS NONE RMXXXX } qty: 1134
    { XAMS NONE RSXXXX } qty: 12
    { XAMS PS   EPXXXX } qty: 1
    { XAMS PS   ERXXXX } qty: 10
    { XAMS WAR  RWSXXX } qty: 219
    { XAMS WAR  RWXXXX } qty: 88405
    { ENXB WAR  RWIXXX } qty: 1
    { ENXB WAR  RWSXXX } qty: 1
    { ENXB WAR  RWXXXX } qty: 49
    { ENXL GO   DBVUCB } qty: 1
    { ENXL GO   DBVUFB } qty: 2
    { ENXL GO   DBXXXX } qty: 14
    { ENXL GO   DYFUXR } qty: 1
    { ENXL NONE DBXXXX } qty: 4
    { ENXL NONE ESXXXX } qty: 18
    { ENXL NONE RMXXXX } qty: 330
    { ENXL WAR  RWSXXX } qty: 6929
    { ENXL WAR  RWXXXX } qty: 7400
    { TNLB NONE ESXXXX } qty: 5
    { MLXB GO   DBXXXX } qty: 4
    { MLXB NONE ESXXXX } qty: 24
    { MLXB NONE RMXXXX } qty: 6
    { ALXP GO   DBXXXX } qty: 40
    { ALXP NONE DBXXXX } qty: 3
    { ALXP NONE DCXXXX } qty: 8
    { ALXP NONE DWXXXX } qty: 1
    { ALXP NONE ESXXXX } qty: 238
    { ALXP NONE RSXXXX } qty: 37
    { ALXP PS   ERXXXX } qty: 1
    { ALXP WAR  RWSXXX } qty: 28
    { ALXB GO   DBXXXX } qty: 13
    { ALXB MTN  DTXXXX } qty: 9
    { ALXB NONE ESXXXX } qty: 13
    { ALXB NONE RMXXXX } qty: 2
    { ALXA GO   DBXXXX } qty: 2
    { ALXA NONE ESXXXX } qty: 11
    { ALXA NONE EUXXXX } qty: 1
    { TNLA NONE ESXXXX } qty: 28
    { misc GO   DBXXXX } qty: 8153
    { misc GO   DYXXXB } qty: 23
    { misc NONE DWXXXX } qty: 2
    { misc NONE ESXXXX } qty: 8581
    { misc NONE EUOGSR } qty: 2
    { misc NONE EUOGXB } qty: 5
    { misc NONE EUOGXR } qty: 2
    { misc NONE EUOIXR } qty: 4                                          (see next page)
```

```
BRANCHES CONTINUED
    { misc NONE EUOMXB } qty: 2
    { misc NONE EUXXXX } qty: 183
    { misc NONE RMXXXX } qty: 1
    { misc PS   ERXXXX } qty: 1
    { misc WAR  RWMXXX } qty: 4998
```

1.2. Types of Instruments

The following sections describe the instruments available on the EURONEXT UTP market data stream:

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

1.2.1. Equities

The sample below illustrates the details of an equity:

```
instr # 81/1193028 = 171062340
    PriceCurrency               string{EUR}
    Symbol                      string{HIPAY}
    Description                 string{HIPAY GROUP}
    SecurityType                string{NONE}
    FOSMarketId                 XPAR
    PriceType                   uint8{2}
    CFICode                     string{ESXXXX}
    CountryOfIssue              string{FRA}
    RoundLot                    float64{1}
    SecuritySubType             string{41}
    InternalCreationDate        Timestamp{2015-07-23 17:36:03:079}
    InternalModificationDate    Timestamp{2015-07-24 04:25:01:096}
    InternalSourceId            uint16{95}
    InternalAggregationId       uint16{95}
    InternalEntitlementId       int32{1034}
    LocalCodeStr                string{FR0012821916}
    ISIN                        string{FR0012821916}
    ICB_SubsectorCode           uint32{2795}
    PriceIncrement_dynamic_TableId      uint32{6226028}
    SecurityTradingId           string{FR0012821916}
    OperatingMIC                string{XPAR}
    MARKET_EURONEXT_InstrumentGroupCode string{16}
    MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice    string{1}
    MARKET_EURONEXT_NominalMarketValueOfTheSecurity     float64{11}
    MARKET_EURONEXT_QuantityNotation    string{UNT}
    MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending      string{0}
    MARKET_EURONEXT_EligibleToPEA       bool{True}
    MARKET_EURONEXT_TypeOfMarketAdmission       char{J}
```

1.2.2. Bonds

The sample below illustrates the details of a bond:

```
instr # 202/759528 = 424384232
    PriceCurrency               string{EUR}
    Symbol                      string{BBBRR}
    Description                 string{BPI FRN 31MAR25}
    SecurityType                string{NONE}
    FOSMarketId                 XLIS
    PriceType                   uint8{1}
    CFICode                     string{DBXXXX}
    CountryOfIssue              string{PRT}
    RoundLot                    float64{100000}
    SecuritySubType             string{259}
    InternalCreationDate        Timestamp{2015-03-27 18:49:10:328}
    InternalModificationDate    Timestamp{2015-07-24 04:25:10:230}
    InternalSourceId            uint16{95}
    InternalAggregationId       uint16{95}
    InternalEntitlementId       int32{1034}
    LocalCodeStr                string{PTBBRROE0048}
    ISIN                        string{PTBBRROE0048}
    ICB_SubsectorCode           uint32{8355}
    PriceIncrement_dynamic_TableId      uint32{6226032}
    SecurityTradingId           string{PTBBRROE0048}
    OperatingMIC                string{XLIS}
    MARKET_EURONEXT_InstrumentGroupCode string{R2}
    MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice    string{2}
    MARKET_EURONEXT_NominalMarketValueOfTheSecurity     float64{1}
    MARKET_EURONEXT_QuantityNotation    string{FMT}
    MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending      string{0}
    MARKET_EURONEXT_TypeOfMarketAdmission       char{G}
```

1.2.3. Rights

The sample below illustrates the details of a right:

```
instr # 265/902381 = 556647661
    PriceCurrency               string{EUR}
    Symbol                      string{DK97B}
    Description                 string{SP500 2240BOC0716B}
    SecurityType                string{NONE}
    StrikePrice                 float64{2240}
    FOSMarketId                 XAMS
    PriceType                   uint8{2}
    CFICode                     string{RMXXXX}
    CountryOfIssue              string{NLD}
    RoundLot                    float64{1}
    SecuritySubType             string{257}
    StrikeCurrency              string{USD}
    InternalCreationDate        Timestamp{2015-06-25 17:32:40:158}
    InternalModificationDate    Timestamp{2015-07-24 04:25:11:593}
    InternalSourceId            uint16{95}
    InternalAggregationId       uint16{95}
    InternalEntitlementId       int32{1034}
    LocalCodeStr                string{NLOO11256979}
    ISIN                        string{NLOO11256979}
    UnderlyingLocalCodeStr      string{US78378X1072}
    PriceIncrement_dynamic_TableId      uint32{6226028}
    SecurityTradingId           string{NLOO11256979}
    OperatingMIC                string{XAMS}
    MARKET_EURONEXT_InstrumentGroupCode string{MI}
    MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice    string{1}
    MARKET_EURONEXT_NominalMarketValueOfTheSecurity     float64{18.86}
    MARKET_EURONEXT_QuantityNotation    string{UNT}
    MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending      string{0}
    MARKET_EURONEXT_TypeOfMarketAdmission       char{G}
```

1.2.4. Warrants

The sample below illustrates the details of a warrant:

```
instr # 265/876341 = 556621621
    PriceCurrency               string{EUR}
    Symbol                      string{WROD1}
    Description                 string{ROODMICROTECWAR1}
    SecurityType                string{WAR}
    StrikePrice                 float64{0.13}
    FOSMarketId                 XAMS
    PriceType                   uint8{2}
    CFICode                     string{RWSXXX}
    CountryOfIssue              string{NLD}
    RoundLot                    float64{1}
    SecuritySubType             string{72}
    InternalCreationDate        Timestamp{2014-10-31 18:34:03:967}
    InternalModificationDate    Timestamp{2015-07-24 04:25:11:593}
    InternalSourceId            uint16{95}
    InternalAggregationId       uint16{95}
    InternalEntitlementId       int32{1034}
    LocalCodeStr                string{NLOO10938130}
    ISIN                        string{NLOO10938130}
    ICB_SubsectorCode           uint32{9576}
    PriceIncrement_dynamic_TableId      uint32{6226028}
    SecurityTradingId           string{NLOO10938130}
    OperatingMIC                string{XAMS}
    MARKET_EURONEXT_InstrumentGroupCode string{K1}
    MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice    string{1}
    MARKET_EURONEXT_NominalMarketValueOfTheSecurity     float64{0}
    MARKET_EURONEXT_QuantityNotation    string{UNT}
    MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending      string{0}
    MARKET_EURONEXT_TypeOfMarketAdmission       char{L}
```

1.2.5. Indices

The sample below illustrates the details of an index:

```
instr # 81/1193081 = 171062393
    PriceCurrency               string{USD}
    Symbol                      string{IU11S}
    Description                 string{EASYETF U11S INAV}
    SecurityType                string{INDEX}
    FOSMarketId                 XPAR
    PriceType                   uint8{2}
    CFICode                     string{MRIXXX}
    CountryOfIssue              string{LUX}
    RoundLot                    float64{1}
    SecuritySubType             string{68}
    InternalCreationDate        Timestamp{2015-07-23 17:36:08:094}
    InternalModificationDate    Timestamp{2015-07-24 04:25:01:238}
    InternalSourceId            uint16{95}
    InternalAggregationId       uint16{95}
    InternalEntitlementId       int32{1035}
    LocalCodeStr                string{NSCFR0IU11S6}
    ISIN                        string{NSCFR0IU11S6}
    PriceIncrement_static       float64{0.01}
    ICB_SubsectorCode           uint32{8995}
    SecurityTradingId           string{NSCFR0IU11S6}
    OperatingMIC                string{XPAR}
    MARKET_EURONEXT_InstrumentGroupCode string{88}
    MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice    string{1}
    MARKET_EURONEXT_NominalMarketValueOfTheSecurity     float64{0}
    MARKET_EURONEXT_QuantityNotation    string{UNT}
    MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending      string{8}
    MARKET_EURONEXT_TypeOfMarketAdmission       char{7}
```

1.3. Specific Referential Tags

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

1.3.1. OperatingMIC & SegmentMIC

The values of the referential tags **OperatingMIC** and **SegmentMIC** conveyed on the EURONEXT UTP 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 2 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	`XAMS`	`XAMS`	NYSE Euronext - Euronext Amsterdam
	`XAMS`	`ALXA`	NYSE Euronext - Alternext Amsterdam
	`XAMS`	`TNLA`	NYSE Euronext - Traded but not listed Amsterdam
	`XBRU`	`XBRU`	NYSE Euronext - Euronext Brussels
	`XBRU`	`ALXB`	NYSE Euronext - Alternext Brussels
	`XBRU`	`MLXB`	NYSE Euronext - Marché Libre Brussels
	`XBRU`	`TNLB`	NYSE Euronext - Trading Facility Brussels
	`XLIS`	`XLIS`	NYSE Euronext - Euronext Lisbon
	`XLIS`	`ENXL`	NYSE Euronext - Easynext Lisbon
	`XPAR`	`XPAR`	NYSE Euronext - Euronext Paris
	`XPAR`	`XMLI`	NYSE Euronext - Marché Libre Paris
	`XPAR`	`ALXP`	NYSE Euronext - Alternext Paris
	`XLUX`	`XLUX`	Luxembourg Stock Exchange

1.3.2. MARKET_EURONEXT_InstrumentGroupCode

The referential tag **MARKET_EURONEXT_InstrumentGroupCode** is disseminated via FeedOS market data stream in *Referential* to detail the code of the Instrument Group.

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

Table 3 MARKET_EURONEXT_InstrumentGroupCode – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_InstrumentGroupCode`	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	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the code of the Instrument Group.

1.3.3. MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice

The referential tag **MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice** is disseminated via FeedOS market data stream in *Referential* to indicate the unit in which the security is quoted.

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

Table 4 MARKET_EURONEXT_TypeOfUnitOfExpressionForInstrumentPrice – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_TypeOfUnitOfExpression ForInstrumentPrice`	FeedOS tag name.
Numeric ID	`11051`	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 unit in which the security is quoted.
Possible Values	`1`	In units.
	`2`	As a % of nominal (excluding accrued interest – *Clean*).
	`3`	As a % of nominal (including accrued interest – *Dirty*).
	`8`	In kilograms.
	`9`	In ounces

1.3.4. MARKET_EURONEXT_NominalMarketValueOfTheSecurity

The referential tag **MARKET_EURONEXT_NominalMarketValueOfTheSecurity** is disseminated via FeedOS market data stream in *Referential* to indicate is the amount of the instrument nominal value.

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

Table 5 MARKET_EURONEXT_NominalMarketValueOfTheSecurity – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_NominalMarketValueOfThe Security`	FeedOS tag name.
Numeric ID	`11052`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the amount of the instruments' nominal value.

1.3.5. MARKET_EURONEXT_QuantityNotation

The referential tag **MARKET_EURONEXT_QuantityNotation** is disseminated via FeedOS market data stream in *Referential* to specify the nature of the amount expression used for negotiating the instrument on the market.

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

Table 6 MARKET_EURONEXT_QuantityNotation – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_QuantityNotation`	FeedOS tag name.
Numeric ID	`11053`	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 nature of the amount expression used for negotiating the instrument on the market.
Possible Values	`UNT`	In unit (i.e. number of shares)
	`FMT`	In facial amount (i.e. bonds expressed in %)
	`Null`	Not applicable

1.3.6. MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending

The referential tag **MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending** is disseminated via FeedOS market data stream in *Referential* to indicate whether the security listed underlies any loan contracts, meaning it has been admitted to the Deferred Settlement system and/or to the lending market.

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

Table 7 MARKET_EURONEXT_IndicatorOfUnderlyingSecurityOnLending – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_IndicatorOfUnderlying SecurityOnLending`	FeedOS tag name.
Numeric ID	`11054`	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying whether the security underlies any loan contracts or not.
Possible Values	0	Instrument neither eligible for the SRD, nor eligible for the Loan and Lending Market.
	1	Instrument eligible for the SRD and for the loan and Lending Market.
	2	Instrument eligible for the SRD long only.
	3	Instrument eligible for the Loan and Lending Market but not for the SRD.
	4	Easy-to-borrow eligible for the SRD and for the Loan and Lending Market.
	8	Non significant.

1.3.7. MARKET_EURONEXT_EligibleToPEA

The referential tag **MARKET_EURONEXT_EligibleToPEA** is disseminated via FeedOS market data stream in *Referential*, to indicate whether the instrument is eligible to a Plan Epargne Action or not.

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

Table 8 MARKET_EURONEXT_EligibleToPEA – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_EligibleToPEA	FeedOS tag name.
Numeric ID	11055	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	Boolean data type.
Format	*[Exchange Specific Value]*	An ***exchange specific value***, indicating whether the instrument is eligible to a Plan Epargne Action or not.
Possible Values	True	Eligible
	False	Non-eligible.

1.3.8. MARKET_EURONEXT_TypeOfMarketAdmission

The values of the referential tag **MARKET_EURONEXT_TypeOfMarketAdmission** conveyed on the EURONEXT UTP market data stream are disseminated via FeedOS data stream in *Referential* to indicate the type of market to which a security has been listed.

FeedOS implementation of the tag MARKET_EURONEXT_TypeOfMarketAdmission is described in the table below:

Table 9 MARKET_EURONEXT_TypeOfMarketAdmission – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_TypeOfMarketAdmission	FeedOS tag name.
Numeric ID	11056	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	Char	Char data type.
Format	*[Exchange Specific Value]*	An ***exchange specific value***, indicating the type of market to which a security has been listed.
Possible Values	A	Instruments traded on the primary market
	B	Instruments traded on the secondary market
	C	Instruments traded on the New Market
	D	Non regulated market / instruments traded on the free market ('Marché Libre')
	E	Non regulated market / Alternext
	F	Non listed
	G	Regulated Market / Non equities
	H	Regulated Market / Equities / Segment A
	I	Regulated Market / Equities / Segment B
	J	Regulated Market / Equities / Segment C
	K	Regulated Market / All securities / Special Segment
	L	Regulated Market / Equities / Other instruments
	S	OPCVM, SICOMI non listed (French Investment Funds)
	6	Off Market
	7	Gold, Currencies, and Indices of Euronext
	9	Foreign

2. Quotation Data

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

2.1. Quotation Values

The examples below shows the possible values of an instrument on the EURONEXT UTP market data stream:

```
InstrumentStatusL1
-- 81/1193028
        BID: 14.86      187     @1
        ASK: 15.18      201     @1
        LastPrice                       float64{14.86}
        LastTradeQty                    float64{9}
        DailyHighPrice                  float64{15.48}
        DailyLowPrice                   float64{14.8}
        DailyTotalVolumeTraded          float64{2524}
        DailyTotalAssetTraded           float64{38089.29}
        LastTradePrice                  float64{14.86}
        LastTradeTimestamp              Timestamp{2015-07-24 12:25:40:154}
        InternalDailyOpenTimestamp      Timestamp{2015-07-24 07:00:00:001}
        InternalDailyHighTimestamp      Timestamp{2015-07-24 07:00:00:001}
        InternalDailyLowTimestamp       Timestamp{2015-07-24 07:30:40:116}
        InternalPriceActivityTimestamp  Timestamp{2015-07-24 12:25:40:155}
        LowLimitPrice                   float64{14.12}
        HighLimitPrice                  float64{15.6}
        TradingStatus                   17=ReadyToTrade
        DailyOpeningPrice               float64{15.48}
        CurrentBusinessDay              Timestamp{2015-07-24}
        PriceActivityMarketTimestamp    Timestamp{2015-07-24 12:25:40:154}
        InternalDailyBusinessDayTimestamp  Timestamp{2015-07-24 07:00:00:001}
        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** conveyed on the EURONEXT UTP 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 10 TradingStatus – technical implementation in FeedOS

Component	Value	Description
Tag Name	`TradingStatus`	FeedOS tag name.
Numeric ID	9100	FeedOS unique ID disseminated on the S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Enum`	Enum data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the characteristics of the trading status.
Possible Values	2	Trading Halt
	15	New 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 EURONEXT UTP 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 EURONEXT UTP market data stream:

- 2.3.1.1. MARKET_EURONEXT_CrossOrderIndicator
- 2.3.1.2. MARKET_EURONEXT_OrderPriorityTimestamp
- 2.3.1.3. MARKET_EURONEXT_TradeOffExchangeFlag
- 2.3.1.4. MARKET_EURONEXT_TradingVenue
- 2.3.1.5. 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 EURONEXT UTP 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 11 MARKET_EURONEXT_CrossOrderIndicator – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_CrossOrderIndicator`	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	*`[Exchange Specific Value]`*	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.
	`7`	The cumulated volume data of the day, either traded on UTP or received from TCS (CumulativeQuantity = Cumulative Quantity of TCS + CumulativeQuantity of UTP).
	`N`	For future use.
	`R`	Retail Matching Facility Trade.

2.3.1.2. MARKET_EURONEXT_OrderPriorityTimestamp

Each time an order is given, the values of the quotation tag **MARKET_EURONEXT_OrderPriorityTimestamp** conveyed on the EURONEXT UTP 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_OrderPriorityTimestamp` is described in the table below:

Table 12 MARKET_EURONEXT_OrderPriorityTimestamp – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_OrderPriorityTimestamp`	FeedOS tag name.
Numeric ID	`15052`	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	`Timestamp`	Timestamp data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, providing the timestamp of the order priority.

2.3.1.3. 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 EURONEXT UTP 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 13 MARKET_EURONEXT_TradeOffExchangeFlag – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_TradeOffExchangeFlag`	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	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing whether the trade relates to a block or a deal negotiated on MiFID rules.
Possible Values	`B`	Block Trade
	`N`	Regular Trade or Negotiated Deal
	`Null`	Undefined

2.3.1.4. MARKET_EURONEXT_TradingVenue

Each time a security is traded on a specific Euronext market, the values of the quotation tag **MARKET_EURONEXT_TradingVenue** conveyed on the EURONEXT UTP 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_TradingVenue` is described in the table below:

Table 14 MARKET_EURONEXT_TradingVenue – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_TradingVenue`	FeedOS tag name.
Numeric ID	`15054`	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 Euronext MIC where the security is traded.
Possible Values	`Initial MIC of the instrument`	If the TCS Operation Type = D, E, H, I.
	`Other Euronext MIC, but different than the initial MIC of the instrument`	If the TCS Operation Type = R.

2.3.1.5. MARKET_EURONEXT_TradeTypeOfOperation

Each time a trade occurs, the values of the quotation tag **MARKET_EURONEXT_TradeTypeOfOperation** conveyed on the EURONEXT UTP 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 values currently available for the tag `MARKET_EURONEXT_TradeTypeOfOperation` is described in the table below:

Table 15 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 / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the type of operation.
Possible Values	`D`	Delta Neutral Liffe Connect
	`E`	Market VWap Operation
	`H`	Out of Market
	`I`	Investment Funds
	`R`	Secondary Listing Place

2.3.2. Other Values

The following subsections describe additional specific quotation tags on the EURONEXT UTP market data stream:

- 2.3.3. InternalDailyClosingPriceType
- 2.3.3.1. MARKET_EURONEXT_HaltReason
- 2.3.3.2. MARKET_EURONEXT_ClassState.

2.3.3. InternalDailyClosingPriceType

The values of the quotation tag **InternalDailyClosingPriceType** conveyed on the EURONEXT UTP 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 16 InternalDailyClosingPriceType – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`InternalDailyClosingPriceType`	FeedOS tag name.
Numeric ID	`9155`	FeedOS unique ID disseminated on S&P Capital IQ Real-Time Solutions data stream. This is the numeric equivalent of the tag name.
Type	`Char`	Char data type.
Format	*`[Internal Specific Value]`*	An ***internal specific value***, detailing the type of daily closing price, as described below.
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).

2.3.3.1. MARKET_EURONEXT_HaltReason

Each time an instrument is halted from trading, the values of the quotation tag **MARKET_EURONEXT_HaltReason** conveyed on the EURONEXT UTP 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 17 MARKET_EURONEXT_HaltReason – technical implementation in FeedOS

Component	Value	Description
Tag Name	`MARKET_EURONEXT_HaltReason`	FeedOS tag name.
Numeric ID	`14550`	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 reason for halting for an instrument.

Table 17 MARKET_EURONEXT_HaltReason – technical implementation in FeedOS (Continued)

Component	Value	Description
Possible Values	0	Not applicable.
	C	Opening or trade price outside dynamic collars.
	K	Knock-Out
	M	Manually halting by market operations.
	R	Halted.
	<Null> or <Space>	Instrument not halted or information not available.

2.3.3.2. 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 EURONEXT UTP 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 values currently available for the tag `MARKET_EURONEXT_ClassState` is described in the table below:

Table 18 MARKET_EURONEXT_ClassState – technical implementation in FeedOS

Component	Value	Description
Tag Name	MARKET_EURONEXT_ClassState	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	*[Exchange Specific Value]*	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 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. 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.

4. Special Behavior

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

4.1. Level1 Market Data Kinematics – Close

In the Level1 Market Data Kinematics **before 2014-10-27**, the CLOSE signal and the Closing Price are being sent at the end of the Continuous Trading Phase, at 17:30 Paris Time, while the `DailyClosingPrice` is being updated at 17:35 Paris Time, as shown below (times expressed as UTC):

```
"TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE  15:29:59:065  170619665   *  *  11.975  8915@5  *  *
TE  15:29:59:072  170619665   *  *  *  *  11.98  4529@7
SI  15:30:00:012  170619665   CLOSE  11.975
TE  15:30:00:012  170619665   11.975  *  *  *  *  *  C
VU  15:30:00:012  170619665   MARKET_EURONEXT_ClassState=CLCA  TradingStatus=15
VU  15:30:00:018  170619665   HighLimitPrice=12.69  LowLimitPrice=11.26
TE  15:30:00:018  170619665   *  *  AT_BEST  22142@1  *  *
TE  15:30:00:018  170619665   *  *  AT_BEST  78142@2  *  *
...
TE  15:35:00:141  170619665   11.97  649  *  *  *  *
VU  15:35:00:142  170619665   DailyOpeningPrice=11.895
VU  15:35:00:142  170619665   DailyClosingPrice=11.97  MARKET_EURONEXT_ClassState=TAL
TradingStatus=17
VU  15:35:00:145  170619665   HighLimitPrice=12.33  LowLimitPrice=11.62
TE  15:37:00:691  170619665   *  *  *  *  11.98  80596@13
TE  15:39:05:380  170619665   *  *  11.975  5165@2  *  *
VU  15:40:00:000  170619665   MARKET_EURONEXT_ClassState=LAMO  TradingStatus=18
VU  15:40:00:001  170619665   HighLimitPrice=12.69  LowLimitPrice=11.26
```

* 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.

In the Level1 Market Data Kinematics **after 2014-10-27**, the CLOSE signal and the Closing Price are being sent after the Auction Phase, at 17:35 Paris Time. The `DailyClosingPrice` is no longer updated and equals the Closing Price, as shown in the example below (times expressed as UTC):

```
"TE (TradeEvent) : MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY BID_PRICE BID_QTY ASK_PRICE
ASK_QTY *CONTENT_MASK* *FLAGS*"
"VU (ValuesUpdate) : SERVER_TIME INSTRUMENT VALUES..."
"SI (TradeEvent) *SIGNAL* : SERVER_TIME INSTRUMENT SIGNAL LAST_PRICE"

TE  15:29:59:036.840  170619665  *  *  *  *  11.94  2900@7
TE  15:30:00:018.541  170619665  *  *  *  *  AT_BEST 325@1
TE  15:30:00:018.541  170619665  *  *  *  *  AT_BEST 18522@2
VU  15:30:00:013.127  170619665  MARKET_EURONEXT_ClassState=CLCA TradingStatus=15
VU  15:30:00:018.384  170619665  HighLimitPrice=12.65  LowLimitPrice=11.22
VU  15:30:00:036.060  170619665  LastAuctionPrice=11.91  LastAuctionVolume=30611
VU  15:30:00:130.998  170619665  LastAuctionVolume=25309
TE  15:30:01:005.902  170619665  *  *  12.53   3401@1  *  *
TE  15:30:02:008.489  170619665  *  *  AT_BEST 15900@1 *  *
TE  15:30:02:009.031  170619665  *  *  AT_BEST 15941@2 *  *
...
VU  15:35:00:138.835  170619665   HighLimitPrice=12.655   LowLimitPrice=11.225
SI  15:35:00:140.002  170619665   CLOSE   11.94
TE  15:35:00:140.002  170619665   11.94  *  *  *  *  *  C
VU  15:35:00:140.002  170619665   InternalDailyClosingPriceType=d
MARKET_EURONEXT_ClassState=TAL  TradingStatus=17
VU  15:35:00:142.943  170619665   HighLimitPrice=12.295   LowLimitPrice=11.585
...
TE  15:39:57:017.634  170619665   *  *  *  *  11.94   19052@2
TE  15:39:57:040.728  170619665   *  *  *  *  11.94   13565@1
VU  15:40:00:003.360  170619665   MARKET_EURONEXT_ClassState=LAMO TradingStatus=18
VU  15:40:00:003.383  170619665   HighLimitPrice=12.655   LowLimitPrice=11.225
```

4.2. Market News – ActionAffectingState

Each time an extraordinary event occurs, the values of the field **ActionAffectingState** conveyed on the EURONEXT UTP market data stream are disseminated via FeedOS data stream in the free flow of the *Market News* to describe the event that caused the change in the stock state:

- in the callback carrying the Level1 event `notif_MarketNews()`, for C++
- in the event handler `MarketNewsEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifMarketNewsEvent`, for Java.

FeedOS implementation of the field `ActionAffectingState` is described below:

Table 19 ActionAffectingState – technical implementation in FeedOS Market News

Component	Value	Description
Field Name	`ActionAffectingState`	Field name in the free flow of the FeedOS Market News.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, describing the event that caused the change in the stock state.

Table 19 ActionAffectingState – technical implementation in FeedOS Market News (Continued)

Component	Value	Description
Possible Values	0	Not applicable.
	B	Trading after knock-out period (beginning of the trading day)
	C	Trading on the instrument at the opening (sent before O)
	D	Cancelled programmed opening
	E	Instrument halted by Market Operations on knock-out
	I	Instrument halted by Issuer/LP on knock-out
	M	Instrument manually halted by Market Operations
	N	Instrument is being initialized (beginning of the trading day)
	O	Instrument opened
	P	Deferred programmed opening
	R	Automatic halting at the class auction
	T	Beginning of a trading after knock-out period
	Y	Beginning of a one side only period
	Z	End of a one side only period
	<Null> or <Space>	Indicates that the 'Order Entry' flag on an instrument has just changed.

Below is an example of the current implementation of the field `ActionAffectingState`:

```
MarketNews
        OrigMarketId    XPAR
        OrigUTCTime     2014-05-19 08:26:14:603
        Headline        StockStateChange
        Content XPAR@FR0011855550|FIXStatus=2|StartDateHalting=null|
        StartTimeHalting=null|ProgOpeningTime=null|OrderEntryRejection=N|
        InstrumentState=H|InstrumentTradingStatus=|HaltReason=R|
        ActionAffectingState=M|InstrumentStateTCS=|PeriodSize= |
        RelatedInstruments      81/1018249
```

4.3. Market News – RequestForSize

Each time the exchange sends a `RequestForSize` message, this is disseminated in the EURONEXT UTP market data stream in *Market News*:

- in the callback carrying the Level1 event `notif_MarketNews()`, for C++
- in the event handler `MarketNewsEventHandler`, for C#
- in the callback carrying the Level1 event `quotNotifMarketNewsEvent`, for Java.

Below is an example showing the `RequestForSize` message dissemination in the Market News:

```
"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..."
“MN : MARKE NEWS”

MN  null  null  XPAR  Normal  RequestForSize | SymbolIndex=22370 | RFSID=5 | Side=1 |
VolumeRequested=500000  related_instruments: 81/754144
```

4.4. Microsecond Timestamp Precision on Level1 Market Data

In the Level1 Market Data disseminated, the server timestamp displays microsecond units, as shown in the example below (highlighted in `green`):

```
TE      11:00:22:091.520     556597898       *       *       *       *       -121.42  1@1
TE      11:00:22:091.612     556597898       *       *       121.75   26@5    *       *
TE      11:00:22:091.612     556597898       *       *       *       *       6       942@39
TE      11:00:22:091.868     556597898       *       *       123.25   23@4    *       *
```

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.