S&P Capital IQ's Real-Time Solutions

QuantFEED® Feed Description

CME Feed

Reference n°: 20131004

S&P Capital IQ's Real-Time Solutions (QuantHouse®) – QuantFEED®
QuantFEED® Feed Description
Reference 20131004
October 04, 2013

Corporate Headquarters
S&P Capital IQ's Real-Time Solutions (QuantHouse®)
52 Rue de la Victoire
75009 Paris
France
Tel: +33 (0) 1 73 02 32 11
Fax: +33 (0) 1 73 02 32 12

UK Office
10 Foster Lane
London EC2V 6HR
United Kingdom
Tel: +44 (0) 203 107 1676

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

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

www.quanthouse.com

TABLE OF CONTENTS

QUANTFEED® CME FEED DESCRIPTION

As part of S&P Capital IQ's Real-Time Solutions's QuantFEED® documentation, this feed description provides you with details about the types of data broadcast on the CME market data stream, their possible values and current QuantFEED® technical implementation for:

Please note that S&P Capital IQ's Real-Time Solutions reproduces CME's instrument codification scheme, which is different from the CFI ISO standard, as shown in the following sections.

Moreover, the exchange does not provide the instruments' description in the market data stream. To obtain it, download and use the following files:

- ftp://ftp.cmegroup.com/fprf/cmeg.opt.prf.xml
- ftp://ftp.cmegroup.com/fprf/cmeg.fut.prf.xml
- ftp://ftp.cmegroup.com/fprf/cmeg.strat.prf.xml.

1. Feed CME GLOBEX

The topics this feed description covers include:

1.1. CME GLOBEX – Referential Data

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

1.1.1. CME GLOBEX – Available Markets and Branches

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

1.1.1.1. CME GLOBEX – Markets

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

Table 1 List of markets available on CME market data stream

QuantFEED® Market ID	Market
XCME	Chicago Mercantile Exchange

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

```
MARKETS
market # 309    CC=US/UNITED STATES OF AMERICA/CHICAGO,DESCR=CHICAGO MERCANTILE EXCHANGE,
WEB=www.cme.com
    MIC = XCME
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
```

1.1.1.2. CME GLOBEX – Branches

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

```
BRANCHES
    { XCME FUT  FCAXSX } qty: 2096
    { XCME FUT  FCXXSX } qty: 18
    { XCME FUT  FFCXSX } qty: 499
    { XCME FUT  FFDXSX } qty: 212
    { XCME FUT  FFIXSX } qty: 125
    { XCME MLEG FMAXSX } qty: 3808
    { XCME MLEG FMCXSX } qty: 260
    { XCME MLEG FMDXSX } qty: 2020
    { XCME MLEG FMIXSX } qty: 88
    { XCME MLEG OCAFPS } qty: 4580
    { XCME MLEG OCEFPS } qty: 4580
    { XCME MLEG OMAFXS } qty: 5266
    { XCME MLEG OMEFXS } qty: 703
    { XCME MLEG OMXFXS } qty: 190
    { XCME MLEG OPAFPS } qty: 4580
    { XCME MLEG OPEFPS } qty: 4580
    { XCME OPT  OCAFPS } qty: 28998
    { XCME OPT  OCEFPS } qty: 11833
    { XCME OPT  OPAFPS } qty: 28998
    { XCME OPT  OPEFPS } qty: 11833
```

1.1.2. Types of Instruments on CME GLOBEX

The following sections illustrate the instruments' characteristics on CME market data stream, according to their type:

- 1.1.2.1. Futures
- 1.1.2.2. Multilegs
- 1.1.2.3. Options.

1.1.2.1. Futures

The sample below illustrates the details of a future:

```
instr # 309/890731 = 648910699
    PriceCurrency               string{USD}
    Symbol                      string{6E}
    SecurityType                string{FUT}
    StdMaturity                 string{201212}
    FOSMarketId                 XCME
    Factor                      float64{125000}
    CFICode                     string{FFCXSX}
    NbLegs                      uint8{0}
    MinTradeVol                 float64{1}
    MatchAlgorithm              string{F}
    InternalCreationDate        Timestamp{2012-09-22 11:32:30:342}
    InternalModificationDate    Timestamp{2012-10-01 21:31:02:715}
    InternalSourceId            uint16{17}
    InternalAggregationId       uint16{17}
    LocalCodeStr                string{6EZ2}
    PriceIncrement_static       float64{0.0001}
    PriceDisplayPrecision       int16{4}
    MaturityYear                uint16{2012}
    MaturityMonth               uint8{12}
```

1.1.2.2. Multilegs

The sample below illustrates the details of a multileg:

```
instr # 309/683342 = 648703310
    PriceCurrency               string{USD}
    Symbol                      string{CUS}
    SecurityType                string{MLEG}
    StdMaturity                 string{201402}
    FOSMarketId                 XCME
    CFICode                     string{FMAXSX}
    NbLegs                      uint8{2}
    MinTradeVol                 float64{1}
    SecuritySubType             string{IS}
    MatchAlgorithm              string{T}
    InternalCreationDate        Timestamp{2012-09-22 11:32:27:696}
    InternalModificationDate    Timestamp{2012-09-30 21:31:10:199}
    InternalSourceId            uint16{17}
    LocalCodeStr                string{CUSG4-LAVG4}
    PriceIncrement_static       float64{0.2}
    PriceDisplayPrecision       int16{2}
    MaturityYear                uint16{2014}
    MaturityMonth               uint8{2}
    LegFOSInstrumentCode        uint32{648702972}
    LegFOSInstrumentCode_1      uint32{648703444}
    LegRatioQty                 float64{1}
    LegRatioQty_1               float64{1}
    LegFIXSide                  '1'=Buy
    LegFIXSide_1                '2'=Sell
```

1.1.2.3. Options

The sample below illustrates the details of an option:

```
instr # 309/541898 = 648561866
    PriceCurrency               string{USD}
    Symbol                      string{VXT}
    SecurityType                string{MLEG}
    FOSMarketId                 XCME
    CFICode                     string{OCEFPS}
    NbLegs                      uint8{2}
    MinTradeVol                 float64{1}
    MatchAlgorithm              string{F}
    InternalCreationDate        Timestamp{2012-09-22 11:32:27:264}
    InternalModificationDate    Timestamp{2012-09-30 21:31:11:671}
    InternalSourceId            uint16{17}
    LocalCodeStr                string{VXTM3_C1240}
    PriceIncrement_static       float64{2.5e-06}
    PriceDisplayPrecision       int16{0}
    UnderlyingFOSMarketId       XCME
    UnderlyingLocalCodeStr      string{6EM3}
    UnderlyingFOSInstrumentCode uint32{648327718}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{6}
    LegFOSInstrumentCode        uint32{648560746}
    LegFOSInstrumentCode_1      uint32{648327718}
    LegRatioQty                 float64{1}
    LegRatioQty_1               float64{1}
    LegFIXSide                  '1'=Buy
    LegFIXSide_1                '2'=Sell
```

1.2. CME GLOBEX – Quotation Data

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

- 1.2.1. CME GLOBEX – Quotation Values
- 1.2.2. CME GLOBEX – Trading Status.

1.2.1. CME GLOBEX – Quotation Values

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

```
InstrumentStatusL1
-- 309/890731
        BID: 1.2895     15      @9
        ASK: 1.2896     77      @17
        LastPrice                       float64{1.2895}
        LastTradeQty                    float64{1}
        DailyHighPrice                  float64{1.2929}
        DailyLowPrice                   float64{1.2894}
        DailyTotalVolumeTraded          float64{23331}
        DailyTotalAssetTraded           float64{30119.2163999992}
        LastTradePrice                  float64{1.2895}
        LastTradeTimestamp              Timestamp{2012-10-02 07:05:47:488}
        InternalDailyOpenTimestamp      Timestamp{2012-10-01 21:45:00:088}
        InternalDailyCloseTimestamp     Timestamp{2012-10-01 21:00:00:028}
        InternalDailyHighTimestamp      Timestamp{2012-10-02 05:41:48:596}
        InternalDailyLowTimestamp       Timestamp{2012-10-01 22:06:53:601}
        InternalPriceActivityTimestamp  Timestamp{2012-10-02 07:05:56:663}
        SettlPriceType                  uint8{1}
        LowLimitPrice                   float64{0}
        TradingStatus                   17=ReadyToTrade
        DailyOpeningPrice               float64{1.2897}
        DailySettlementPrice            float64{1.2896}
        PreviousDailyTotalVolumeTraded  float64{279741}
        PreviousDailyTotalAssetTraded   float64{360899.086799949}
        PreviousDailyClosingPrice       float64{1.29}
        PreviousBusinessDay             Timestamp{2012-10-01}
        CurrentBusinessDay              Timestamp{2012-10-02}
        PreviousDailySettlementPrice    float64{1.2896}
```

For more details about the fields and tags available in quotation data type, and their possible values, see *FeedOS™ Quotation Tags Guide*.

1.2.2. CME GLOBEX – Trading Status

Each time a modification of the trading status occurs, the values of the quotation tag **Trading Status** conveyed on the CME market data stream are disseminated via QuantFEED®'s 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.

QuantFEED®'s implementation of the tag `TradingStatus` is described in the following table:

Table 2 TradingStatus – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingStatus`	QuantFEED® tag name.
Numeric ID	`9100`	QuantFEED® unique ID disseminated on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`Enum`	Enum data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the characteristics of the trading status.
Possible Values	`2`	Trading Halt
	`5`	Price Indication
	`17`	Ready to Trade
	`18`	Not Available for Trading
	`21`	Pre-Open

2. Feed CME CBOT

The topics this feed description covers include:

- 2.1. CME CBOT – Referential Data
- 2.2. CME CBOT – Quotation Data.

2.1. CME CBOT – Referential Data

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

- 2.1.1. CME CBOT – Available Markets and Branches
- 2.1.2. Types of Instruments on CME CBOT.

2.1.1. CME CBOT – Available Markets and Branches

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

- 2.1.1.1. CME CBOT – Markets
- 2.1.1.2. CME CBOT – Branches.

2.1.1.1. CME CBOT – Markets

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

Table 3 List of markets available on CME market data stream

QuantFEED® Market ID	Market
XCBT	Chicago Board of Trade
XKBT	Kansas City Board of Trade
XMGE	Minneapolis Grain Exchange

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

```
MARKETS
market # 305    CC=US/UNITED STATES OF AMERICA/CHICAGO,DESCR=CHICAGO BOARD OF TRADE,
WEB=www.cbot.com
    MIC = XCBT
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 320    CC=US/UNITED STATES OF AMERICA/KANSAS CITY,DESCR=KANSAS CITY BOARD OF TRADE,
WEB=www.kcbt.com
    MIC = XKBT
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 321    CC=US/UNITED STATES OF AMERICA/MINNEAPOLIS,DESCR=MINNEAPOLIS GRAIN EXCHANGE,
WEB=www.mgex.com
    MIC = XMGE
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
```

2.1.1.2. CME CBOT – Branches

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

```
BRANCHES
    { XCBT FUT  FCAXSX } qty: 269
    { XCBT FUT  FCMXSX } qty: 40
    { XCBT FUT  FFDXSX } qty: 91
    { XCBT FUT  FFIXSX } qty: 20
    { XCBT MLEG FMAXSX } qty: 1844
    { XCBT MLEG FMDXSX } qty: 1104
    { XCBT MLEG FMIXSX } qty: 48
    { XCBT MLEG FMMXSX } qty: 915
    { XCBT MLEG OMAFXS } qty: 50434
    { XCBT MLEG OMEFXS } qty: 8
    { XCBT MLEG OMXFXS } qty: 7475
    { XCBT OPT  OCAFPS } qty: 30271
    { XCBT OPT  OCEFPS } qty: 5427
    { XCBT OPT  OPAFPS } qty: 30268
    { XCBT OPT  OPEFPS } qty: 5429
    { XKBT FUT  FCAXSX } qty: 11
    { XKBT MLEG FMAXSX } qty: 70
    { XKBT MLEG OMAFXS } qty: 261
    { XKBT MLEG OMXFXS } qty: 1
    { XKBT OPT  OCAFPS } qty: 184
    { XKBT OPT  OPAFPS } qty: 169
    { XMGE FUT  FCAXSX } qty: 164
    { XMGE MLEG FMAXSX } qty: 1986
    { XMGE MLEG OMAFXS } qty: 44
    { XMGE MLEG OMXFXS } qty: 2
    { XMGE OPT  OCAFPS } qty: 2038
    { XMGE OPT  OPAFPS } qty: 2040
```

2.1.2. Types of Instruments on CME CBOT

The following sections illustrate the instruments' characteristics on CME market data stream, according to their type:

- 2.1.2.1. Futures
- 2.1.2.2. Multilegs
- 2.1.2.3. Options.

2.1.2.1. Futures

The sample below illustrates the details of a future:

```
instr # 305/256192 = 639887552
    PriceCurrency               string{USD}
    Symbol                      string{ZC}
    SecurityType                string{FUT}
    StdMaturity                 string{201212}
    FOSMarketId                 XCBT
    Factor                      float64{5000}
    CFICode                     string{FCAXSX}
    NbLegs                      uint8{0}
    MinTradeVol                 float64{1}
    MatchAlgorithm              string{K}
    InternalCreationDate        Timestamp{2012-03-15 16:26:57:468}
    InternalModificationDate    Timestamp{2012-10-01 21:31:05:481}
    InternalSourceId            uint16{16}
    InternalAggregationId       uint16{16}
    LocalCodeStr                string{ZCZ2}
    PriceIncrement_static       float64{0.25}
    PriceDisplayPrecision       int16{0}
    MaturityYear                uint16{2012}
    MaturityMonth               uint8{12}
    MARKET_CME_DisplayPricePrimaryDenominator   uint16{8}
    MARKET_CME_DisplayPriceNbOfDecimal  uint16{1}
```

2.1.2.2. Multilegs

The sample below illustrates the details of a multileg:

```
instr # 305/345563 = 639976923
    PriceCurrency               string{USD}
    Symbol                      string{ZR}
    SecurityType                string{MLEG}
    StdMaturity                 string{201207}
    FOSMarketId                 XCBT
    CFICode                     string{FMAXSX}
    NbLegs                      uint8{2}
    MinTradeVol                 float64{1}
    SecuritySubType             string{SP}
    MatchAlgorithm              string{K}
    InternalCreationDate        Timestamp{2012-03-15 16:26:56:450}
    InternalModificationDate    Timestamp{2012-07-13 21:31:07:720}
    InternalSourceId            uint16{16}
    LocalCodeStr                string{ZRN2-ZRU2}
    PriceIncrement_static       float64{0.005}
    PriceDisplayPrecision       int16{3}
    MaturityYear                uint16{2012}
    MaturityMonth               uint8{7}
    LegFOSInstrumentCode        uint32{639668129}
    LegFOSInstrumentCode_1      uint32{639976922}
    LegRatioQty                 float64{1}
    LegRatioQty_1               float64{1}
    LegFIXSide                  '1'=Buy
    LegFIXSide_1                '2'=Sell
```

2.1.2.3. Options

The sample below illustrates the details of an option:

```
instr # 305/540571 = 640171931
    PriceCurrency               string{USD}
    Symbol                      string{OZL}
    SecurityType                string{OPT}
    StrikePrice                 float64{0.05}
    FOSMarketId                 XCBT
    CFICode                     string{OCAFPS}
    NbLegs                      uint8{0}
    MinTradeVol                 float64{1}
    StrikeCurrency              string{USD}
    MatchAlgorithm              string{O}
    InternalCreationDate        Timestamp{2012-05-26 21:31:00:883}
    InternalModificationDate    Timestamp{2012-09-30 21:31:18:078}
    InternalSourceId            uint16{16}
    LocalCodeStr                string{OZLU3_C0500}
    PriceIncrement_static       float64{0.005}
    PriceDisplayPrecision       int16{3}
    UnderlyingFOSMarketId       XCBT
    UnderlyingLocalCodeStr      string{ZLU3}
    UnderlyingFOSInstrumentCode uint32{640578805}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{9}
```

2.2. CME CBOT – Quotation Data

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

- 2.2.1. CME CBOT – Quotation Values
- 2.2.2. CME CBOT – Trading Status.

2.2.1. CME CBOT – Quotation Values

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

```
InstrumentStatusL1
-- 305/256192
        BID: 758.75     1       @1
        ASK: 759        2       @1
        LastPrice                       float64{759}
        LastTradeQty                    float64{1}
        DailyHighPrice                  float64{760}
        DailyLowPrice                   float64{752.75}
        DailyTotalVolumeTraded          float64{6895}
        DailyTotalAssetTraded           float64{5215078.5}
        LastTradePrice                  float64{759}
        LastTradeTimestamp              Timestamp{2012-10-02 07:07:03:440}
        InternalDailyOpenTimestamp      Timestamp{2012-10-01 21:45:00:052}
        InternalDailyCloseTimestamp     Timestamp{2012-10-01 21:00:00:018}
        InternalDailyHighTimestamp      Timestamp{2012-10-02 06:34:56:286}
        InternalDailyLowTimestamp       Timestamp{2012-10-01 22:32:43:792}
        InternalPriceActivityTimestamp  Timestamp{2012-10-02 07:07:11:432}
        SettlPriceType                  uint8{1}
        LowLimitPrice                   float64{716.75}
        HighLimitPrice                  float64{796.75}
        TradingStatus                   17=ReadyToTrade
        DailyOpeningPrice               float64{756.5}
        DailySettlementPrice            float64{756.75}
        PreviousDailyTotalVolumeTraded  float64{156891}
        PreviousDailyTotalAssetTraded   float64{119029918}
        PreviousDailyClosingPrice       float64{757}
        PreviousBusinessDay             Timestamp{2012-09-30}
        CurrentBusinessDay              Timestamp{2012-10-01}
        PreviousDailySettlementPrice    float64{756.75}
        MARKET_CME_PreliminarySettlementPrice   float64{570}
```

For more details about the fields and tags available in quotation data type, and their possible values, see *FeedOS™ Quotation Tags Guide*.

2.2.2. CME CBOT – Trading Status

Each time a modification of the trading status occurs, the values of the quotation tag **Trading Status** conveyed on the CME market data stream are disseminated via QuantFEED®'s 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.

QuantFEED®'s implementation of the tag `TradingStatus` is described in the following table:

Table 4 TradingStatus – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingStatus`	QuantFEED® tag name.
Numeric ID	9100	QuantFEED® unique ID disseminated on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`Enum`	Enum data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the characteristics of the trading status.
Possible Values	2	Trading Halt
	5	Price Indication
	17	Ready to Trade
	18	Not Available for Trading
	21	Pre-Open

3. Feed CME NYMEX

The topics this feed description covers include:

- 3.1. CME NYMEX – Referential Data
- 3.2. CME NYMEX – Quotation Data.

3.1. CME NYMEX – Referential Data

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

- 3.1.1. CME NYMEX – Available Markets and Branches
- 3.1.2. Types of Instruments on CME NYMEX.

3.1.1. CME NYMEX – Available Markets and Branches

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

- 3.1.1.1. CME NYMEX – Markets
- 3.1.1.2. CME NYMEX – Branches.

3.1.1.1. CME NYMEX – Markets

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

Table 5 List of markets available on CME market data stream

QuantFEED® Market ID	Market
XCEC	Commodities Exchange Center
XNYM	New York Mercantile Exchange

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

```
MARKETS
market # 326    CC=US/UNITED STATES OF AMERICA/NEW YORK,DESCR=COMMODITIES EXCHANGE CENTER,
WEB=www.nymex.com
    MIC = XCEC
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
market # 335    CC=US/UNITED STATES OF AMERICA/NEW YORK,DESCR=NEW YORK MERCANTILE EXCHANGE,
WEB=www.nymex.com
    MIC = XNYM
    TimeZone =
    Country =
    NbMaxInstruments = 1000000
```

3.1.1.2. CME NYMEX – Branches

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

```
BRANCHES
    { XCEC FUT  FXXXSX } qty: 186
    { XCEC MLEG FMXXSX } qty: 2433
    { XCEC MLEG OMAFXS } qty: 11441
    { XCEC MLEG OMXFXS } qty: 206
    { XCEC OPT  OCAXPS } qty: 16747
    { XCEC OPT  OCEXPS } qty: 6006
    { XCEC OPT  OPAXPS } qty: 16748
    { XCEC OPT  OPEXPS } qty: 6006
    { XNYM FUT  FCAXSX } qty: 58
    { XNYM FUT  FCMXSX } qty: 8993
    { XNYM FUT  FCXXSX } qty: 16
    { XNYM FUT  FXXXSX } qty: 72
    { XNYM MLEG FMAXSX } qty: 231
    { XNYM MLEG FMMXSX } qty: 34434
    { XNYM MLEG FMXXSX } qty: 575
    { XNYM MLEG OMAFXS } qty: 12494
    { XNYM MLEG OMEFXS } qty: 21
    { XNYM MLEG OMXFXS } qty: 110
    { XNYM OPT  OCAFPS } qty: 34980
    { XNYM OPT  OCEFPS } qty: 40157
    { XNYM OPT  OPAFPS } qty: 34971
    { XNYM OPT  OPEFPS } qty: 40131
```

3.1.2. Types of Instruments on CME NYMEX

The following sections illustrate the instruments' characteristics on CME market data stream, according to their type:

- 3.1.2.1. Futures
- 3.1.2.2. Multilegs
- 3.1.2.3. Options.

3.1.2.1. Futures

The sample below illustrates the details of a future:

```
instr # 335/10975 = 702556895
    PriceCurrency               string{USD}
    Symbol                      string{CL}
    SecurityType                string{FUT}
    StdMaturity                 string{201211}
    FOSMarketId                 XNYM
    Factor                      float64{1000}
    CFICode                     string{FCMXSX}
    NbLegs                      uint8{0}
    MinTradeVol                 float64{1}
    MatchAlgorithm              string{F}
    InternalCreationDate        Timestamp{2012-03-15 17:02:55:942}
    InternalModificationDate    Timestamp{2012-10-01 21:31:08:036}
    InternalSourceId            uint16{15}
    InternalAggregationId       uint16{15}
    LocalCodeStr                string{CLX2}
    PriceIncrement_static       float64{0.01}
    PriceDisplayPrecision       int16{2}
    MaturityYear                uint16{2012}
    MaturityMonth               uint8{11}
```

3.1.2.2. Multilegs

The sample below illustrates the details of a multileg:

```
instr # 335/577929 = 703123849
    PriceCurrency               string{USD}
    Symbol                      string{MB}
    SecurityType                string{MLEG}
    StdMaturity                 string{201212}
    FOSMarketId                 XNYM
    CFICode                     string{FMMXSX}
    NbLegs                      uint8{2}
    MinTradeVol                 float64{1}
    SecuritySubType             string{SP}
    MatchAlgorithm              string{F}
    InternalCreationDate        Timestamp{2012-03-15 17:04:13:328}
    InternalModificationDate    Timestamp{2012-09-30 21:31:10:263}
    InternalSourceId            uint16{15}
    LocalCodeStr                string{MBZ2-MBZ3}
    PriceIncrement_static       float64{0.01}
    PriceDisplayPrecision       int16{2}
    MaturityYear                uint16{2012}
    MaturityMonth               uint8{12}
    LegFOSInstrumentCode        uint32{703017831}
    LegFOSInstrumentCode_1      uint32{703123885}
    LegRatioQty                 float64{1}
    LegRatioQty_1               float64{1}
    LegFIXSide                  '1'=Buy
    LegFIXSide_1                '2'=Sell
```

3.1.2.3. Options

The sample below illustrates the details of an option:

```
instr # 335/568997 = 703114917
    PriceCurrency               string{USD}
    Symbol                      string{LNE}
    SecurityType                string{OPT}
    StrikePrice                 float64{0.425}
    FOSMarketId                 XNYM
    CFICode                     string{OCEFPS}
    NbLegs                      uint8{0}
    MinTradeVol                 float64{1}
    StrikeCurrency              string{USD}
    MatchAlgorithm              string{F}
    InternalCreationDate        Timestamp{2012-03-15 17:03:55:503}
    InternalModificationDate    Timestamp{2012-09-30 21:31:01:105}
    InternalSourceId            uint16{15}
    LocalCodeStr                string{LNEM3_C4250}
    PriceIncrement_static       float64{0.0001}
    PriceDisplayPrecision       int16{4}
    UnderlyingFOSMarketId       XNYM
    UnderlyingLocalCodeStr      string{NGM3}
    UnderlyingFOSInstrumentCode uint32{702548587}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{6}
```

3.2. CME NYMEX – Quotation Data

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

- 3.2.1. CME NYMEX – Quotation Values
- 3.2.2. CME NYMEX – Trading Status.

3.2.1. CME NYMEX – Quotation Values

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

```
InstrumentStatusL1
-- 335/10975
        BID: 92.81      5       @4
        ASK: 92.82      5       @4
        LastPrice                       float64{92.82}
        LastTradeQty                    float64{2}
        DailyHighPrice                  float64{92.82}
        DailyLowPrice                   float64{92.04}
        DailyTotalVolumeTraded          float64{10116}
        DailyTotalAssetTraded           float64{934751.659999993}
        LastTradePrice                  float64{92.82}
        LastTradeTimestamp              Timestamp{2012-10-02 08:21:51:470}
        InternalDailyOpenTimestamp      Timestamp{2012-10-01 21:51:12:180}
        InternalDailyCloseTimestamp     Timestamp{2012-10-01 21:15:00:032}
        InternalDailyHighTimestamp      Timestamp{2012-10-02 08:21:50:153}
        InternalDailyLowTimestamp       Timestamp{2012-10-02 07:12:03:440}
        InternalPriceActivityTimestamp  Timestamp{2012-10-02 08:21:52:216}
        SettlPriceType                  uint8{1}
        LowLimitPrice                   float64{82.48}
        HighLimitPrice                  float64{102.48}
        TradingStatus                   17=ReadyToTrade
        DailyOpeningPrice               float64{92.41}
        DailySettlementPrice            float64{92.48}
        PreviousDailyTotalVolumeTraded  float64{197187}
        PreviousDailyTotalAssetTraded   float64{18235892.6100009}
        PreviousDailyClosingPrice       float64{92.38}
        PreviousBusinessDay             Timestamp{2012-10-01}
        CurrentBusinessDay              Timestamp{2012-10-02}
        PreviousDailySettlementPrice    float64{92.48}
        MARKET_CME_PreliminarySettlementPrice   float64{105.83}
```

For more details about the fields and tags available in quotation data type, and their possible values, see *FeedOS™ Quotation Tags Guide.*

3.2.2. CME NYMEX – Trading Status

Each time a modification of the trading status occurs, the values of the quotation tag **Trading Status** conveyed on the CME market data stream are disseminated via QuantFEED®'s 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.

QuantFEED®'s implementation of the tag `TradingStatus` is described in the following table:

Table 6 TradingStatus – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingStatus`	QuantFEED® tag name.
Numeric ID	9100	QuantFEED® unique ID disseminated on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`Enum`	Enum data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the characteristics of the trading status.
Possible Values	2	Trading Halt
	5	Price Indication
	17	Ready to Trade
	18	Not Available for Trading
	21	Pre-Open

4. Specific Referential Tags

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

- 4.1. Display Price Primary Denominator
- 4.2. Display Price Secondary Denominator
- 4.3. Display Price Number of Decimals.

4.1. Display Price Primary Denominator

The referential tag **Display Price Primary Denominator** is disseminated via S&P Capital IQ's Real-Time Solutions's market data stream in *Referential* to detail the price denominator.

QuantFEED® implementation of the tag `MARKET_CME_DisplayPricePrimaryDenominator` is described in the table below:

Table 7 MARKET_CME_DisplayPricePrimaryDenominator – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_CME_DisplayPricePrimaryDenominator`	QuantFEED® tag name.
Numeric ID	11500	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`UInt16`	UInt16 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the price denominator.

4.2. Display Price Secondary Denominator

The referential tag **Display Price Secondary Denominator** is disseminated via S&P Capital IQ's Real-Time Solutions's market data stream in *Referential* to detail the price numerator.

QuantFEED® implementation of the tag `MARKET_CME_DisplayPriceSecondaryDenominator` is described in the table below:

Table 8 MARKET_CME_DisplayPriceSecondaryDenominator – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_CME_DisplayPriceSecondaryDenominator`	QuantFEED® tag name.
Numeric ID	`11501`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`UInt16`	UInt16 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the price numerator.

4.3. Display Price Number of Decimals

The referential tag **Display Price Number of Decimals** is disseminated via S&P Capital IQ's Real-Time Solutions's market data stream in *Referential*, to indicate the number of decimals the price displays.

QuantFEED® implementation of the tag `MARKET_CME_DisplayPriceNbOfDecimal` is described in the table below:

Table 9 MARKET_CME_DisplayPriceNbOfDecimal – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_CME_DisplayPriceNbOfDecimal`	QuantFEED® tag name.
Numeric ID	`11502`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`UInt16`	UInt16 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the number of decimals the price displays.

5. Specific Quotation Tags

The following sections describe additional, specific quotation tags available on CME market data stream:

- 5.1. Trade Conditions
- 5.2. Other Values.

5.1. Trade Conditions

The following subsections describe the trade conditions on CME market data stream:

- 5.1.1. Trade Condition
- 5.1.2. Match Event Indicator.

5.1.1. Trade Condition

Each time a trade occurs, the values of the quotation tag **Trade Condition** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Context* to detail the conditions of the trade:

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

QuantFEED® implementation of the tag `TradeCondition` is described in the table below:

Table 10 TradeCondition – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradeCondition`	QuantFEED® tag name.
Numeric ID	`277`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the particular condition applicable to the trade.
Possible Values	`E`	Opening Trade
	`1`	Price calculated by CME GLOBEX (implied trade)

5.1.2. Match Event Indicator

The values of the quotation tag **Match Event Indicator** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Context* to identify the beginning or the end of a CME GLOBEX event:

- 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.

QuantFEED® implementation of the tag `MARKET_CME_MatchEventIndicator` is described in the table below:

Table 11 MARKET_CME_MatchEventIndicator – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_CME_MatchEventIndicator`	QuantFEED® tag name.
Numeric ID	`15101`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's 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 beginning or the end of a CME GLOBEX event. If there is no value present, then the message is not at the beginning or the end of a CME GLOBEX event.
Possible Values	`1`	Beginning of the CME GLOBEX event
	`2`	End of the CME GLOBEX event **Note:** 2 will become a valid value in a future release.

5.2. Other Values

The following subsections describe the other values of specific quotation tags on the CME market data stream:

- 5.2.1. Preliminary Settlement Price
- 5.2.2. Settlement Price Type
- 5.2.3. Low Limit Price
- 5.2.4. High Limit Price
- 5.2.5. Daily Total Volume Traded.

5.2.1. Preliminary Settlement Price

The values of the quotation tag **Preliminary Settlement Price** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Other Values* to indicate the preliminary 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.

QuantFEED® implementation of the tag `MARKET_CME_PreliminarySettlementPrice` is described in the table below:

Table 12 MARKET_CME_PreliminarySettlementPrice – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_CME_PreliminarySettlementPrice`	QuantFEED® tag name.
Numeric ID	14740	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, indicating the value of the preliminary settlement price.

5.2.2. Settlement Price Type

The values of the quotation tag **Settlement Price Type** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Other Values* to identify the type of 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.

QuantFEED® implementation of the tag `SettlPriceType` is described in the table below:

Table 13 SettlPriceType – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`SettlPriceType`	QuantFEED® tag name.
Numeric ID	731	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's data stream. This is the numeric equivalent of the tag name.
Type	`UInt8`	UInt8 data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the type of settlement price, as described below.
Possible Values	1	Final
	2	Theoretical
	100	Actual Preliminary Settlement Price *OR* Rounded Preliminary for instruments subject to settlement rounding
	101	Rounded Preliminary Settlement Price

5.2.3. Low Limit Price

The values of the quotation tag **Low Limit Price** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Other Values* to detail the Lower Price threshold of an instrument:

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

- in the callback carrying the Level1 event `quotNotifTradeEventExt`, for Java.

Please notice that any order submitted with prices below this lower limit is rejected.

QuantFEED® implementation of the tag `LowLimitPrice` is described in the table below:

Table 14 LowLimitPrice – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`LowLimitPrice`	QuantFEED® tag name.
Numeric ID	`1148`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's 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 Lower Price threshold.

5.2.4. High Limit Price

The values of the quotation tag **High Limit Price** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Other Values* to detail the Upper Price threshold of an instrument:

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

Please notice that any order submitted with prices above this upper limit is rejected.

QuantFEED® implementation of the tag `HighLimitPrice` is described in the table below:

Table 15 HighLimitPrice – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`HighLimitPrice`	QuantFEED® tag name.
Numeric ID	`1149`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's 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 Upper Price threshold.

5.2.5. Daily Total Volume Traded

The values of the quotation tag **Daily Total Volume Traded** conveyed on the CME market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Other Values* to detail the volume for legs of spreads with prices calculated by CME GOLBEX:

- 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.

Please notice that the volume of spreads' legs is included in the calculation of the total session trade volume.

QuantFEED® implementation of the tag `DailyTotalVolumeTraded` is described in the table below:

Table 16 DailyTotalVolumeTraded – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`DailyTotalVolumeTraded`	QuantFEED® tag name.
Numeric ID	`9130`	QuantFEED® unique ID broadcast on S&P Capital IQ's Real-Time Solutions's 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***, including the volume for legs of spreads.

6. Special Behavior

The following sections detail CME market data stream special behavior:

6.1. Processing the L1 Market Data

At the network level, the exchange sends a UDP Datagram, which contains several messages. The main message – *IncrementalRefresh* – contains several **Market Data Entries** that carry a Bid, an Ask or a Trade.

If the exchange sends several Ask or Bid updates for a given instrument in a single message, the S&P Capital IQ's Real-Time Solutions's Subscription Server ignores the intermediary updates and sends only the last one to prevent the dissemination of outdated prices in high frequency algorithmic trading.

Nevertheless, this **update mechanism is bypassed when**:

- **A trade occurs** – otherwise some of the trades would appear as if happening outside the Bid-Ask Spread. In this case, the latest (updated) quotes – if any – are sent along with the trade.
- **The message processing is completed**. To ensure the best low latency, S&P Capital IQ's Real-Time Solutions does not try to optimize the processing across the messages of a single datagram.

Please notice that the quotes being ignored correspond to the canceled orders or to the changes of the existing orders, not to the trades.

6.2. Resetting the Order Book

If on a specific channel the order books are corrupted on the exchange side, the exchange sends an *Order Book Reset* message, which empties all the order books of the impacted channel. Subsequently, a new order book is created.

6.3. Converting Decimal Price into Fractional

To convert a CME decimal price to a fractional one, use the C++ utility function `convert_CME_decimal_to_fractional_display_price`, as shown in the example below:

```
namespace FeedOS {
/**
* Convert a decimal price into its representation in fractional using the fractional
specification referential.
* Note: this feature is specific to the CME DMA market.
* @param displayPricePrimaryDenominator tdftpd -
*          the input tick Display Format Type for the Primary Denominator
** @param displayPriceSecondaryDenominator tdftsd -
*          the input tick Display Format Type for the Secondary Denominator
** @param displayPriceNbOfDecimal nddp -
*          the input number of decimal in display price
** @param decimal_price the input decimal price
** @param fractional_display_price the output display price in fractional
** @return true if convertion suceeded, false in case of error
**/
  bool convert_CME_decimal_to_fractional_display_price (
     unsigned int displayPricePrimaryDenominator,
     unsigned int displayPriceSecondaryDenominator,
     unsigned int displayPriceNbOfDecimal,
     double decimal_price,
     std::string & fractional_display_price
);
}
```

7. Official Closing Price

On the market CME, the closing price is provided by the market. If the price is not sent, the last trade is used instead. When a stock splits, the closing price is adjusted after the closing. The settlement price is handled when provided by the market.

8. 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: support@quanthouse.com
- Web: http://support.quanthouse.com.