QuantHouse® FeedOS™

FeedOS™ Developer's Notice

SWX Data Feed Migration to SMR3

Reference n°: 20121207

Effective as of: 10 December 2012

Action required from users: Optional

QuantHouse® FeedOS™
FeedOS™ Developer's Notice
Reference 20121207
December 07, 2012

Corporate Headquarters
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

SWX DATA FEED MIGRATION TO SMR3

To reflect the changes caused by the migration of the SWX market data stream to the SMR3 format, QuantHouse® has decided to enhance the content of QuantFEED®.

This developer's notice contains late-breaking information about the implementation of this modification in your applications, which may not be included otherwise in the published documentation. The topics this notice covers include:

1. Update Summary

Table 1 Current update summary

Notice Reference	20121207
Scope	Reference Data
Exchanges	SWX
Effective Date	**2012-12-10**
Impact	• Update of the Quotation Context Tags • FeedOS™ API Upgrade for Feed Replay
Action required	Optional

2. Functional Description

Starting Monday, **December 10, 2012**, QuantHouse® introduces a new referential tag `CCP_Eligible` (**NumericID:** `9552`, **Type:** `Bool`) to accommodate the information disseminated on SWX, following the migration to the SMR3 format

Moreover, the content of the quotation tag `TradingStatus` (**NumericID**: `9100`, **Type:** `Enum`) and specific quotation context tag `MARKET_SWX_TradeTypeIndicator` (**NumericID:** `15450`, **Type:** `String`) changes.

Furthermore, the new quotation context tag `TradeID` (**NumericID:** `1003`, **Type:** `String`) disseminated in QuantFEED®'s Level 1 Data Stream to identify the trade, will be **available upon request only** for QuantHouse® customers using *a dedicated SWX feed handler*.

Also, please be aware that the market specific quotation context tag `MARKET_SWX_TradeOffExchangeFlag` (**NumericID:** `15452`, **Type:** `String`) will be deprecated and no longer used in an upcoming release of the SWX market data stream.

3. QuantFEED® Technical Implementation

The following sections describe the technical implementation of the new or updated quotation context tags:

3.1. CCP Eligible

Each times there is a central counterparty which acts as the buyer to every seller and the seller to every buyer, thus guaranteeing the contractual performance for a security, the values of the referential tag **CCP Eligible** conveyed on the SWX market data stream are disseminated via QuantHouse®'s data stream in *Referential*:

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

Table 2 CCP_Eligible – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`CCP_Eligible`	QuantFEED® tag name.
Numeric ID	`9552`	QuantFEED® unique ID broadcast on QuantHouse®'s data stream. This is the numeric equivalent of the tag name.
Type	`Bool`	Bool data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, indicating that a central counterparty acts as the buyer to every seller and the seller to every buyer, thus guaranteeing the contractual performance for a security.
Possible Values	`True`	CCP Eligible
	`False`	Not CCP Eligible

Below is an example of the current implementation of the referential tags in the SWX market data stream:

```
instr # 256/510485 = 537381397
    PriceCurrency               string{CHF}
    Symbol                      string{LOGN}
    Issuer                      string{Logitech}
    Description                 string{LOGITECH N}
    SecurityType                string{NONE}
    FOSMarketId                 XSWX
    PriceType                   uint8{2}
    CFICode                     string{ESXXXR}
    RoundLot                    float64{1}
    MinTradeVol                 float64{0}
    SecuritySubType             string{Registered Share}
    DatedDate                   Timestamp{2012-09-24}
    MarketSegmentID             string{HS}
    MarketSegmentDesc           string{Main Market}
    InternalCreationDate        Timestamp{2012-12-05 16:17:29:755}
    InternalModificationDate    Timestamp{2012-12-06 01:00:03:228}
    InternalSourceId            uint16{41}
    LocalCodeStr                string{CH0025751329_CHF}
    ISIN                        string{CH0025751329}
    Telekurs_Valor              string{2575132}
    PriceIncrement_dynamic_TableId      uint32{2687086}
    CCP_Eligible                bool{True}
    MARKET_SWX_IssuerCountry    string{CH}
    MARKET_SWX_TradingSessionID string{ABck}
    MARKET_SWX_ListingStateCode string{LI}
    MARKET_SWX_ListingStateDesc string{Listed}
```

3.2. Trading Status

Each time a modification of the trading status occurs, the values of the quotation tag **Trading Status** conveyed on the SWX 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 3 TradingStatus – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradingStatus`	QuantFEED® tag name.
Numeric ID	9100	QuantFEED® unique ID disseminated on QuantHouse®'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.

Table 3 TradingStatus – technical implementation in QuantFEED® (Continued)

Component	Value	Description
Possible Values	2	Trading Halt
	5	Price Indication
	17	Ready to Trade
	18	Not Available for Trading
	21	Pre-Open

Note	For an accurate value of the trading status, you should also process the values disseminated by the following market specific tags: `MARKET_SWX_BookCondition` (**NumericID:** `14452`, **Type:** `Int32`), `MARKET_SWX_SecurityTradingStatus` (**NumericID:** `14453`, **Type:** `Int32`) and `MARKET_SWX_TradingSessionSubID` (**NumericID:** `14454`, **Type:** `String`). For more details about the implementation and possible values of these tags, see *SWX Feed Description*.

3.3. Trade Type Indicator

Each time a trade with a special price occurs, the values of the quotation context tag **Trade Type Indicator** conveyed on the SWX market data stream are disseminated via QuantHouse®'s data stream in *Context* to detail the type of 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 `MARKET_SWX_TradeTypeIndicator` is described in the table below:

Table 4 MARKET_SWX_TradeTypeIndicator – technical implementation in QuantFEED®

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

3.4. Trade ID (Optional)

Each time a trade occurs, the values of the quotation context tag **Trade ID** conveyed on the SWX market data stream are disseminated via QuantHouse®'s data stream in *Context* only for QuantHouse® customers using a dedicated SWX feed handler to identify the trade:

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

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

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

Table 5 TradeID – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`TradeID`	QuantFEED® tag name.
Numeric ID	`1003`	QuantFEED® unique ID broadcast on QuantHouse®'s data stream. This is the numeric equivalent of the tag name.
Type	`String`	String data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, identifying the trade.

Below is an example of the previous and current implementation of the quotation context tags in SWX market data stream:

SWX Old Version

```
InstrumentStatusL1
-- 298/500015
        BID: 20.85      1890    @4
        ASK: 20.87      7457    @12
        LastPrice                       float64{20.85}
        LastTradeQty                    float64{200}
        DailyHighPrice                  float64{21.07}
        DailyLowPrice                   float64{20.8}
        DailyTotalVolumeTraded          float64{1721990}
        DailyTotalAssetTraded           float64{36036313.4}
        LastTradePrice                  float64{20.85}
        LastTradeTimestamp              Timestamp{2012-10-29 09:53:55:245}
        InternalDailyOpenTimestamp      Timestamp{2012-10-29 08:01:05:600}
        InternalDailyCloseTimestamp     Timestamp{2012-10-26 15:31:23:237}
        InternalDailyHighTimestamp      Timestamp{2012-10-29 08:01:52:550}
        InternalDailyLowTimestamp       Timestamp{2012-10-29 09:00:37:690}
        InternalPriceActivityTimestamp  Timestamp{2012-10-29 09:54:04:117}
        TradingStatus                   17=ReadyToTrade
        LastOffBookTradePrice           float64{20.83}
        LastOffBookTradeQty             float64{240}
        LastOffBookTradeTimestamp       Timestamp{2012-10-29 09:10:08}
        DailyOpeningPrice               float64{21.03}
        PreviousDailyTotalVolumeTraded  float64{5677343}
        PreviousDailyTotalAssetTraded   float64{120134904.46}
        PreviousDailyClosingPrice       float64{21.16}
        PreviousBusinessDay             Timestamp{2012-10-26}
        CurrentBusinessDay              Timestamp{2012-10-29}
        LastAuctionPrice                float64{21.03}
        LastAuctionVolume               float64{81927}
        DailyTotalOffBookVolumeTraded   float64{1440}
        DailyTotalOffBookAssetTraded    float64{29984.81005}
        InternalLastAuctionTimestamp    Timestamp{2012-10-29 08:00:58:694}
        MARKET_SWX_BookCondition        int32{3}
        MARKET_SWX_SecurityTradingStatus int32{17}
        MARKET_SWX_TradingSessionSubID  string{2}

EV 298/500015           2012-10-29 10:02:12:566 /ServerUTCTime: 2012-10-29 10:02:12:754
content: LastPrice LastTradeQty Context
        LastTradeQty  = 97
        LastPrice     = 20.89

CONTEXT:
        TradeConditionsDictionaryKey:   uint32{20971621}
        MARKET_SWX_TradeTypeIndicator:  NX
```

SWX New Version

```
InstrumentStatusL1
-- 298/500015
        BID: 17.77      9514    @1
        ASK: 17.79      6786    @4
        LastPrice                       float64{17.79}
        LastTradeQty                    float64{361}
        DailyHighPrice                  float64{35}
        DailyLowPrice                   float64{17}
        DailyTotalVolumeTraded          float64{550364}
        DailyTotalAssetTraded           float64{9772479.15}
        LastTradePrice                  float64{17.79}
        LastTradeTimestamp              Timestamp{2012-10-29 09:42:24:666}
        InternalDailyOpenTimestamp      Timestamp{2012-10-29 07:27:03:251}
        InternalDailyCloseTimestamp     Timestamp{2012-10-26 20:31:52:008}
        InternalDailyHighTimestamp      Timestamp{2012-10-29 07:27:03:251}
        InternalDailyLowTimestamp       Timestamp{2012-10-29 08:06:44:918}
        InternalPriceActivityTimestamp  Timestamp{2012-10-29 09:42:24:836}
        TradingStatus                   17=ReadyToTrade
        LastOffBookTradePrice           float64{10}
        LastOffBookTradeQty             float64{1000}
        LastOffBookTradeTimestamp       Timestamp{2012-10-28 23:00:00}
        DailyOpeningPrice               float64{35}
        PreviousDailyTotalVolumeTraded  float64{2935069}
        PreviousDailyTotalAssetTraded   float64{52422217.5299998}
        PreviousDailyClosingPrice       float64{18.18}
        PreviousBusinessDay             Timestamp{2012-10-26}
        CurrentBusinessDay              Timestamp{2012-10-29}
        LastAuctionPrice                float64{17.75}
        LastAuctionVolume               float64{299940}
        DailyTotalOffBookVolumeTraded   float64{1004633}
        DailyTotalOffBookAssetTraded    float64{17848937.1}
        InternalLastAuctionTimestamp    Timestamp{2012-10-29 09:00:52:412}
        MARKET_SWX_BookCondition        int32{3}
        MARKET_SWX_SecurityTradingStatus int32{17}
        MARKET_SWX_TradingSessionSubID  string{2}

EV 298/500015           2012-10-28 23:00:00     /ServerUTCTime: 2012-10-29 09:31:13:903
content: LastPrice LastTradeQty OffBookTrade Context
        LastTradeQty  = 1000
        LastPrice     = 10

CONTEXT:
        TradeID:                        OB20121029000050
        TradeConditionsDictionaryKey:   uint32{83886180}
        MARKET_SWX_TradeTypeIndicator:  30
```

4. Upgrading FeedOS™ API for Replay Purposes

To be able to replay the SWX Data Feed that is recorded after the migration date – **December 10, 2012** –, you should upgrade the FeedOS™ API to the minimum required version, as described in the table below:

Table 6 Currently required version to replay SWX Data Feed

Language	FeedOS™ API – minimum required version
C++	3.6.3.3
C#	2.4.3.4

For more details about the upgrade procedure, see *FeedOS™API Guide*.

5. Finding the Latest Information

For the latest documentation and product updates, additional support and training, please contact our support services:

- E-mail: support@quanthouse.com
- Web: http://support.quanthouse.com.