S&P Capital IQ's Real-Time Solutions

QuantFEED® Feed Description

ICE Feed

Reference n°: 20130816

S&P CAPITAL IQ
McGRAW HILL FINANCIAL

S&P Capital IQ's Real-Time Solutions (QuantHouse®) – QuantFEED®
QuantFEED® Feed Description
Reference 20130816
August 16, 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® ICE 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 ICE market data stream, their possible values and current QuantFEED® technical implementation.

The topics this feed description covers include:

1. Referential Data

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

1.1. Available Markets and Branches

This section details the list of Markets and Branches available on ICE market data stream.

1.1.1. Markets

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

Table 1 List of markets available on ICE market data stream

QuantFEED® Market ID	Market
IEPA	Intercontinental Exchange Ltd. **Note:** This MIC has been introduced to accommodate data about the OTC instruments broadcast on a possible upcoming version of ICE market data stream. Currently, S&P Capital IQ's Real-Time Solutions does not disseminate any instrument details associated with this MIC.
ICEU	Intercontinental Exchange – ICE Futures Europe
IFCA	Intercontinental Exchange – ICE Futures Canada
ICUS	Intercontinental Exchange United States

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

```
MARKETS
market # 290    CC=GB/UNITED KINGDOM/LONDON,DESCR=INTERCONTINENTAL EXCHANGE LTD.,
WEB=www.intcx.com
    MIC = IEPA
    TimeZone = America/New_York
    Country = US
    NbMaxInstruments = 2000000
market # 432    CC=UK/UNITED KINGDOM/LONDON,DESCR=INTERCONTINENTAL EXCHANGE - ICE FUTURES
EUROPE, WEB=www.theice.com
    MIC = ICEU
    TimeZone = Europe/London
    Country = UK
    NbMaxInstruments = 2000000
market # 471    CC=CA/CANADA/WINNIPEG,DESCR=ICE FUTURES CANADA, WEB=www.theice.com
    MIC = IFCA
    TimeZone = America/Winnipeg
    Country = US
    NbMaxInstruments = 2000000
market # 507    CC=US/UNITED STATES OF AMERICA/NEW YORK,DESCR=ICE FUTURES U.S. INC,
WEB=www.theice.com
    MIC = ICUS
    TimeZone = America/New_York
    Country = US
    NbMaxInstruments = 2000000
```

1.1.2. Branches

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

```
BRANCHES
    { ICEU FUT  FCXXXX } qty: 5102
    { ICEU FUT  MXXXXX } qty: 60
    { ICEU MLEG MRXXXX } qty: 4727
    { ICEU OPT  OCAXXX } qty: 19696
    { ICEU OPT  OCEXXX } qty: 12688
    { ICEU OPT  OPAXXX } qty: 19696
    { ICEU OPT  OPEXXX } qty: 12688
    { IFCA FUT  FCXXXX } qty: 56
    { IFCA MLEG MRXXXX } qty: 219
    { IFCA OPT  OCAXXX } qty: 3006
    { IFCA OPT  OPAXXX } qty: 3006
    { ICUS FUT  FCXXXX } qty: 584
    { ICUS FUT  MXXXXX } qty: 92
    { ICUS INDEX MRIXXX } qty: 24
    { ICUS MLEG MRXXXX } qty: 2496
    { ICUS OPT  OCAXXX } qty: 22670
    { ICUS OPT  OPAXXX } qty: 22670
```

1.2. Types of Instruments

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

Please note that some of the **multileg instruments** do not have a full description of the leg ratios, as the exchange message does not disseminate this type of information. For more details, please refer to the Intercontinental Exchange Web site to identify the ratio of the crack spread and other strategy types.

1.2.1. Futures

The sample below illustrates the details of a future:

```
instr # 507/750280 = 1064006344
    PriceCurrency               string{USX}
    Symbol                      string{CT}
    Description                 string{Cotton No. 2 TAS - NYCC - Mar14}
    SecurityType                string{FUT}
    StdMaturity                 string{201403}
    FOSMarketId                 ICUS
    Factor                      float64{50000}
    ContractMultiplier          float64{50000}
    CFICode                     string{MXXXXX}
    SecuritySubType             string{TradeAtSettlement}
    ProductComplex              string{CT    20140221}
    InternalCreationDate        Timestamp{2013-07-03 09:11:07:520}
    InternalModificationDate    Timestamp{2013-07-08 11:32:50:054}
    InternalSourceId            uint16{40}
    LocalCodeStr                string{820735}
    PriceIncrement_static       float64{0.01}
    MaturityYear                uint16{2014}
    MaturityMonth               uint8{2}
    MaturityDay                 uint8{21}
    MARKET_ICE_ContractSymbol   string{CT  FMH0014_Z}
```

1.2.2. Index

The sample below illustrates the details of an index:

```
instr # 507/775439 = 1064031503
    PriceCurrency               string{USD}
    Symbol                      string{RUSS}
    Description                 string{Russell Top 200 - Index - Cash}
    SecurityType                string{INDEX}
    StdMaturity                 string{201307}
    FOSMarketId                 ICUS
    CFICode                     string{MRIXXX}
    SecuritySubType             string{Index}
    ProductComplex              string{RUSS  20130715}
    InternalCreationDate        Timestamp{2013-07-15 12:11:18:421}
    InternalModificationDate    Timestamp{2013-07-15 13:21:52:743}
    InternalSourceId            uint16{40}
    LocalCodeStr                string{1280027}
    PriceIncrement_static       float64{0.01}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{7}
    MaturityDay                 uint8{15}
    MARKET_ICE_ContractSymbol   string{RUSSFDN1513_IRQH}
```

1.2.3. Multilegs

The sample below illustrates the details of a multileg:

```
instr # 507/753471 = 1064009535
    SecurityType                string{MLEG}
    FOSMarketId                 ICUS
    CFICode                     string{MRXXXX}
    InternalCreationDate        Timestamp{2013-07-08 13:31:16:279}
    InternalModificationDate    Timestamp{2013-07-08 13:31:16:279}
    InternalSourceId            uint16{40}
    LocalCodeStr                string{90724151}
    PriceIncrement_static       float64{0.01}
    UnderlyingFOSMarketId       ICUS
    UnderlyingLocalCodeStr      string{800539}
    UnderlyingFOSInstrumentCode uint32{1064006178}
    LegRatioQty                 float64{1}
    LegRatioQty_1               float64{1}
    LegFIXSide                  '1'=Buy
    LegFIXSide_1                '1'=Buy
    MARKET_ICE_ContractSymbol   string{}
```

1.2.4. Options/Futures Trade at Settlement

The sample below illustrates the details of an option/future trade at settlement:

```
instr # 507/774775 = 1064030839
    PriceCurrency               string{USD}
    Symbol                      string{KEO}
    Description                 string{EUR/USD 125 TAS - NYCC - Sep13}
    SecurityType                string{FUT}
    StdMaturity                 string{201309}
    FOSMarketId                 ICUS
    Factor                      float64{125000}
    ContractMultiplier          float64{125000}
    CFICode                     string{MXXXXX}
    SecuritySubType             string{TradeAtSettlement}
    ProductComplex              string{KEO   20130916}
    InternalCreationDate        Timestamp{2013-07-15 12:11:18:085}
    InternalModificationDate    Timestamp{2013-07-15 13:21:52:409}
    InternalSourceId            uint16{40}
    LocalCodeStr                string{2072329}
    PriceIncrement_static       float64{5e-05}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{9}
    MaturityDay                 uint8{16}
    MARKET_ICE_ContractSymbol   string{KEO FMU0013_Z}
```

1.2.5. Options

The sample below illustrates the details of an option:

```
instr # 432/757456 = 906727120
    PriceCurrency               string{USD}
    Symbol                      string{WBS}
    Description                 string{WTI Crude Futures - WTI - Sep13}
    SecurityType                string{OPT}
    StdMaturity                 string{201309}
    StrikePrice                 float64{97.25}
    FOSMarketId                 ICEU
    CFICode                     string{OCAXXX}
    SecuritySubType             string{Option on Future/Month}
    ProductComplex              string{WBS   20130815C00097250}
    InternalCreationDate        Timestamp{2013-07-08 12:58:25:202}
    InternalModificationDate    Timestamp{2013-07-08 12:58:25:202}
    InternalSourceId            uint16{188}
    LocalCodeStr                string{90942384}
    PriceIncrement_static       float64{0.01}
    UnderlyingFOSMarketId       ICEU
    UnderlyingLocalCodeStr      string{461133}
    UnderlyingFOSInstrumentCode uint32{906723363}
    MaturityYear                uint16{2013}
    MaturityMonth               uint8{8}
    MaturityDay                 uint8{15}
    MARKET_ICE_ContractSymbol   string{WBS FMU0013_OMCA0000097252081513}
```

1.3. Specific Referential Tags

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

1.3.1. Contract Symbol

The referential tag **Contract Symbol** is disseminated via S&P Capital IQ's Real-Time Solutions's market data stream in *Referential* to detail the symbol of the security.

Caution	Please note that in the current version of the ICE feed handler, the data disseminated via the tag `LocalCodeStr` is the ICE trading code sent directly by the market. In the previous version of the feed, these values were disseminated via the tag `InternalMagic_0`. Moreover, the values previously disseminated via the tag `LocalCodeStore` are now broadcast over the tag `MARKET_ICE_ContractSymbol`, which is a market-specific tag. In the previous version of the feed, S&P Capital IQ's Real-Time Solutions replaced all spaces to underscores (_) and removed all exclamation marks (`!`). Currently, the data disseminated via the tag `MARKET_ICE_ContractSymbol` is no longer altered by S&P Capital IQ's Real-Time Solutions.

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

Table 2 MARKET_ICE_ContractSymbol – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_ContractSymbol`	QuantFEED® tag name.
Numeric ID	`11600`	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	`String`	Strings data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the symbol of the security.

2. Quotation Data

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

- 2.1. Quotation Values
- 2.2. Trading Status
- 2.3. Specific Quotation Tags.

2.1. Quotation Values

The example below shows the possible values of an instrument on ICE market data stream:

```
InstrumentStatusL1
-- 432/750228
        BID: 60.3       10
        ASK: 60.5       50
        LastPrice                       float64{60.3}
        LastTradeQty                    float64{10}
        DailyHighPrice                  float64{60.3}
        DailyLowPrice                   float64{59.9}
        DailyTotalVolumeTraded          float64{90}
        DailyTotalAssetTraded           float64{5403.7}
        LastTradePrice                  float64{60.3}
        LastTradeTimestamp              Timestamp{2012-08-09 13:44:43:486}
        InternalDailyOpenTimestamp      Timestamp{2012-08-09 08:26:25:916}
        InternalDailyCloseTimestamp     Timestamp{2012-08-08 23:00:00:997}
        InternalDailyHighTimestamp      Timestamp{2012-08-09 13:44:43:487}
        InternalDailyLowTimestamp       Timestamp{2012-08-09 08:26:25:916}
        InternalPriceActivityTimestamp  Timestamp{2012-08-09 14:08:00:532}
        TradingStatus                   17=ReadyToTrade
        LastOffBookTradePrice           float64{57.85}
        LastOffBookTradeQty             float64{25}
        LastOffBookTradeTimestamp       Timestamp{2012-07-17 13:25:07}
        SessionVWAPPrice                float64{60.025}
        DailyOpeningPrice               float64{59.9}
        PreviousDailyTotalVolumeTraded  float64{260}
        PreviousDailyTotalAssetTraded   float64{15631.9}
        PreviousDailyClosingPrice       float64{60.3}
        PreviousBusinessDay             Timestamp{2012-08-08}
        CurrentBusinessDay              Timestamp{2012-08-09}
        PreviousDailySettlementPrice    float64{60.258}
        LastAuctionPrice                float64{60.27}
        DailyTotalOffBookVolumeTraded   float64{0}
        DailyTotalOffBookAssetTraded    float64{0}
        OpenInterest                    float64{3.21}
        InternalLastAuctionTimestamp    Timestamp{2012-08-09 06:00:00:036}
        MARKET_ICE_BlockVolume          float64{25}
        MARKET_ICE_EFPVolume            float64{100}
```

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

2.2. Trading Status

Each time a modification of the trading status occurs, the values of the quotation tag **Trading Status** conveyed on the ICE 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 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
	`21`	Pre-Open

2.3. Specific Quotation Tags

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

- 2.3.1. Trade Conditions
- 2.3.2. Other Specific Values.

2.3.1. Trade Conditions

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

- 2.3.1.1. Aggressor Side
- 2.3.1.2. Block Trade Type
- 2.3.1.3. System Priced Leg Type.

2.3.1.1. Aggressor Side

The values of the quotation tags **Aggressor Side** conveyed on the ICE market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's data stream in *Context* to specify if the aggressor is a buyer or a seller:

- 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 `AggressorSide` is described in the table below:

Table 4 AggressorSide – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`AggressorSide`	QuantFEED® tag name.
Numeric ID	`9356`	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	`Char`	Char data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, identifying the side of the aggressor.
Possible Values	`Space`	No aggressor
	`1`	Buy Side
	`2`	Sell Side

2.3.1.2. Block Trade Type

Each time a block trade occurs, the values of the quotation tags **Block Trade Type** conveyed on the ICE market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's 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.

QuantFEED® implementation of the tag `MARKET_ICE_BlockTradeType` is described in the table below. Please note that all the trades having a trade condition are flagged as off-book trades:

Table 5 MARKET_ICE_BlockTradeType – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_BlockTradeType`	QuantFEED® tag name.
Numeric ID	`15501`	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 characteristics of a block trade.
Possible Values	`K`	Block Trade
	`E`	EFP Trade
	`S`	EFS Trade
	`V`	Bilateral O-Exchange Trade
	`O`	NG EFP/EFS Trade
	`9`	CCX EFP Trade
	`J`	EFR Trade

2.3.1.3. System Priced Leg Type

A **System Priced Leg** is the result of a trade in the outright spread. The underlying leg markets spread trades display the last price. Additionally, this outright spread trade is not attributed to any derived legs into or out of the spread.

Caution	Do not include the outright trades in the high/low/close prices, as they are flagged as off-book trades.

Thus, each time a trade in the outright spread occurs, the values of the quotation tags **System Priced Leg Type** conveyed on the ICE market data stream are disseminated via S&P Capital IQ's Real-Time Solutions's 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.

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

Table 6 MARKET_ICE_SystemPricedLegType – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_SystemPricedLegType`	QuantFEED® tag name.
Numeric ID	`15502`	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 type of system priced leg.
Possible Values	`C`	System Priced Crack Spread Leg
	`S`	System Priced Leg

2.3.2. Other Specific Values

The following subsections detail other specific quotation tags available on ICE market data stream:

- 2.3.2.1. Block Volume
- 2.3.2.2. EFS Volume
- 2.3.2.3. EFP Volume
- 2.3.2.4. Interval Price Limits on Hold.

2.3.2.1. Block Volume

Each time a block trade occurs, the values of the quotation tag **Block Volume** conveyed on the ICE 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 `MARKET_ICE_BlockVolume` is described in the following table:

Table 7 MARKET_ICE_BlockVolume – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_BlockVolume`	QuantFEED® tag name.
Numeric ID	`14500`	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	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the volume of a block trade.

2.3.2.2. EFS Volume

Each time a swap exchange occurs, the values of the quotation tag **EFS Volume** conveyed on the ICE 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 `MARKET_ICE_EFSVolume` is described in the following table:

Table 8 MARKET_ICE_EFSVolume – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_EFSVolume`	QuantFEED® tag name.
Numeric ID	`14501`	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	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the volume of the exchanged swaps.

2.3.2.3. EFP Volume

Each time a physical exchange occurs, the values of the quotation tag **EFP Volume** conveyed on the ICE 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 `MARKET_ICE_EFPVolume` is described in the following table:

Table 9 MARKET_ICE_EFPVolume – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_EFPVolume`	QuantFEED® tag name.
Numeric ID	`14502`	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	`Float64`	Float64 data type.
Format / Possible Values	*`[Exchange Specific Value]`*	An ***exchange specific value***, detailing the volume of the exchanged physicals.

2.3.2.4. Interval Price Limits on Hold

Interval Price Limits (IPL) purpose is to avoid sudden movements (in both directions) in the market during a short period of time. If IPL is violated, a *Hold period* is instituted to prevent prices outside of IPL range. Subsequent IPL notifications are sent out to market participants about this period (*IPL Hold Start – IPL Hold End*).

Thus, each time the price limits are exceeded, the values of the quotation tag **Interval Price Limits on Hold** conveyed on the ICE 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 `MARKET_ICE_IntervalPriceLimitsOnHold` is described in the following table:

Table 10 MARKET_ICE_IntervalPriceLimitsOnHold – technical implementation in QuantFEED®

Component	Value	Description
Tag Name	`MARKET_ICE_IntervalPriceLimitsOnHold`	QuantFEED® tag name.
Numeric ID	`14503`	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	`Bool`	Boolean data type.
Format	*`[Exchange Specific Value]`*	An ***exchange specific value***, specifying if a Hold period is instituted, following an IPL violation.
Possible Values	`Space / Empty`	Under normal trading circumstances, there are no values.
	`True`	Interval Price Limit was violated and a Hold period is instituted.
	`Reset`	Interval Price Limit is no longer on Hold.

3. Special Behavior

S&P Capital IQ's Real-Time Solutions merge the implied prices/quantities provided by the Intercontinental Exchange with the outright prices/quantity.

4. Official Closing Price

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

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