Tick-by-Tick Data

Tick-by-tick data corresponding to the data shown in the TWS Time & Sales Window is available starting with TWS v969 and API v973.04. The maximum number of simultaneous tick-by-tick subscriptions allowed for a user is determined by the same formula used to calculate maximum number of market depth subscriptions **Limitations**.

- The tick type field is case sensitive - it must be BidAsk, Last, AllLast, MidPoint. AllLast has additional trade types such as combos, derivatives, and average price trades which are not included in Last.
- Tick-by-tick data for options is currently only available historically and not in real time.
- Tick-by-tick data for indices is only provided for indices which are on CME.
- Tick-by-tick data is not available for combos.
- No more than 1 tick-by-tick request can be made for the same instrument within 15 seconds.

```
            client.reqTickByTickData(19001, ContractSamples.USStockAtSmart(), "Last", 0, false);
            client.reqTickByTickData(19002, ContractSamples.USStockAtSmart(), "AllLast", 0, false);
            client.reqTickByTickData(19003, ContractSamples.USStockAtSmart(), "BidAsk", 0, true);
            client.reqTickByTickData(19004, ContractSamples.EurGbpFx(), "MidPoint", 0, false);
```

Depending on the data type chosen in **IBApi::EClient::reqTickByTickData**, data will be returned to one of the functions **IBApi::EWrapper::tickByTickAllLast**, **IBApi::EWrapper::tickByTickBidAsk**, or **IBApi::EWrapper::tickByTickMidPoint**. Additionally, if a non-zero value is input for the argument *numberOfTicks* in **IBApi::EClient::reqTickByTickData**, historical tick data is first returned to one of the functions **IBApi::EWrapper::historicalTicksLast**, **IBApi::EWrapper::historicalTicksBidAsk**, or **IBApi::EWrapper::historicalTicks**, respectively.

```
        public void tickByTickBidAsk(int reqId, long time, double bidPrice, double askPrice, decimal
        bidSize, decimal askSize, TickAttribBidAsk tickAttribBidAsk)
        {
            Console.WriteLine("Tick-By-Tick. Request Id: {0}, TickType: BidAsk, Time: {1}, BidPrice:
        {2}, AskPrice: {3}, BidSize: {4}, AskSize: {5}, BidPastLow: {6}, AskPastHigh: {7}",
                reqId, Util.UnixSecondsToString(time, "yyyyMMdd-HH:mm:ss"),
        Util.DoubleMaxString(bidPrice), Util.DoubleMaxString(askPrice),
        Util.DecimalMaxString(bidSize), Util.DecimalMaxString(askSize), tickAttribBidAsk.BidPastLow,
        tickAttribBidAsk.AskPastHigh);
        }
```

```
        public void tickByTickAllLast(int reqId, int tickType, long time, double price, decimal
        size, TickAttribLast tickAttribLast, string exchange, string specialConditions)
        {
            Console.WriteLine("Tick-By-Tick. Request Id: {0}, TickType: {1}, Time: {2}, Price: {3},
        Size: {4}, Exchange: {5}, Special Conditions: {6}, PastLimit: {7}, Unreported: {8}",
                reqId, tickType == 1 ? "Last" : "AllLast", Util.UnixSecondsToString(time, "yyyyMMdd-
        HH:mm:ss"), Util.DoubleMaxString(price), Util.DecimalMaxString(size), exchange,
        specialConditions, tickAttribLast.PastLimit, tickAttribLast.Unreported);
        }
```

- Trade data for non-reportable trades, such as combos and block trades, are included in the 'AllLast' data type available with the real time data feed but are not recorded in the historical database.

```
        public void tickByTickMidPoint(int reqId, long time, double midPoint)
        {
            Console.WriteLine("Tick-By-Tick. Request Id: {0}, TickType: MidPoint, Time: {1},
        MidPoint: {2}",
```

```
                reqId, Util.UnixSecondsToString(time, "yyyyMMdd-HH:mm:ss"),
        Util.DoubleMaxString(midPoint));
         }
```

Halted and Unhalted ticks

Effective with **TWS 10.15+** and **TWS API 10.15+** a new Tick-By-Tick attribute has been introduced. The tick attribute *pastLimit* is also returned with historical Tick-By-Tick responses. Check **Historical Halted and Unhalted ticks** section.

- If tick has zero price, zero size and pastLimit flag is set - this is "Halted" tick.
- If tick has zero price, zero size and followed immediately after "Halted" tick - this is "Unhalted" tick.

Canceling Tick-By-Tick Subscriptions

Tick-by-tick subscriptions can be cancelled using the function **IBApi::EClient::cancelTickByTickData**

```
            client.cancelTickByTickData(19001);
            client.cancelTickByTickData(19002);
            client.cancelTickByTickData(19003);
            client.cancelTickByTickData(19004);
```