

## API Release Notes - Build 9.66

The modifications below are in **build 9.66** of the TWS API. Please note that all the enhancements for release 9.66 require the minimum server version 54 (TWS version 919), except where otherwise noted.

For clarification on any of the items listed, refer to the appropriate section in the API Reference Guide, or send us an email at API Support.

### Enhancements

#### Opt Out of Default SmartRouting for ASX-Directed Orders

ASX exchange regulations require that SmartRouting be selected by default for orders routed directly to ASX, and require that any opting-out of SmartRouting be done on an order by order basis.

You can opt out of default SmartRouting for orders routed directly to ASX by using a new boolean Order attribute **optOutSmartRouting**. This new attribute defaults to false unless explicitly set to true. When **optOutSmartRouting** is set to false, orders routed directly to ASX will NOT use SmartRouting. When the attribute is set to true, orders routed directly to ASX orders WILL use SmartRouting.

When developing your own custom API application, you must not allow orders to be routed directly to ASX by default, and you must require opting out of SmartRouting on an order by order basis. For more information on these regulations, see the ASX website.

This enhancement applies to all APIs and requires TWS server version 56 or higher.

#### Support for Frozen Market Data

The API can now receive frozen market data from Trader Workstation. Frozen market data is the last data recorded in our system. To receive frozen market data, use the new method **reqMarketDataType( )** with the *type* parameter set to "Frozen." The new method looks like this:

**reqMarketDataType**(int *type*) request

where *type* = 1 for real-time streaming market data or 2 for frozen market data.

TWS sends a **marketDataType**(*type*) callback to the API, where *type* is set to Frozen or RealTime, to announce that market data has been switched between frozen and real-time. This notification occurs only when market data switches between real-time and frozen. The **marketDataType( )** callback accepts a reqId parameter and is sent per every subscription because different contracts can generally trade on a different schedule.

During normal trading hours, the API receives real-time market data. If you use the **reqMarketDataType(frozen)** call, you are telling TWS to automatically switch to frozen market data after the close. Then, before the opening of the next trading day, market data will automatically switch back to real-time market data.

This enhancement applies to all socket-based and ActiveX APIs and requires TWS server version 55 or higher.

## Support for Hedging Orders

You can now add hedge orders using the API. Hedge orders are child orders that take additional fields. There are four types of hedging orders supported by the API:

- » Delta
- » Beta
- » FX
- » Pair

To support these orders, the following fields have been added to order properties:

- » **hedgeType**: Type = string; possible values = D, B, F or P (corresponding to the four types of hedging orders listed above); attribute is HedgingType.
- » **hedgeParam**: Type = string; value depends on the hedgeType; sent from the API only if hedgeType is NOT null. hedgeParam is required for Pair hedge order, optional for Beta hedge orders, and ignored for Delta and FX hedge orders.

The following rules apply:

- » All hedge orders must have a parent order submitted first. The hedge order will provide the parent order ID.
- » If the hedgeType is Beta, the beta sent in the hedgeParam can be zero, which means it is not used.
- » Delta is only valid if the parent order is an option and the child order is a stock.

This enhancement applies to the ActiveX, C++ and Java APIs and requires TWS server version 54 or higher.

## Support for Smart Combo Routing

You can now add smartComboRoutingParams to an order. **smartComboRoutingParams** is similar to AlgoParams in that it makes use of tag/value pairs to add parameters to combo orders. The new parameters cover the following capabilities:

- » **Priority** - User can specify which leg to be executed first.  
Tag = **LegInPrio**  
Values = -1, 0 or 1
- » **Discretionary Amount** - When one leg is executed, we can adjust the other leg by up to a discretionary amount.  
Tag = **MaxSegSize**  
Value = An amount
- » **Market-If-Touched Timeout** - For Market-If-Touched combo orders, we record the firstTradeTime of the first fill of the first leg to execute, and the lastTradeTime of the last partial fill. For these kinds of orders, you can now specify timeout values of the last fill and the timeout since the first fill, in seconds.  
Tags = **ChangeToMktTime1** is the timeout after the last fill, and **ChangeToMktTime2** is the timeout after the first fill.  
Value = Number of seconds
- » **Market-If-Touched Stop-Loss** - Specify an absolute stop-loss amount per combo. If specified and if the implied execution price of the combo (based on a leg that has already been executed and current market data) exceeds the combo price plus the stop-loss amount, we convert the order from LMT to MKT immediately in order to finish executing the combo order. If the stop-loss amount is specified but timeouts have not been specified, we will continue to try to execute the second leg at the calculated LMT price until it either executes or the stop-loss amount is reached.  
Tag = **ChangeToMktOffset**  
Value = An amount.
- » **Maximum Leg-In Size** - Specify the maximum allowed leg-in size per segment.  
Tag = **MaxSegSize**  
Value = Unit of combo size

- » **Discretionary Percentage** = Specify a percentage of the combo price. This applies to scale combos in which the discretionary amount is calculated from the current scale level. When the discretionary amount is entered as a percentage, the API will convert that to a dollar amount according to the combo. This amount will be updated when the order price changes or for scale orders for each level. You can enter a value for this parameter or for the Discretionary Amt extended attribute one at a time, but not both at the same time.

Tag = **DiscretionaryPct**

Value = A value between 0 and 100.

This enhancement applies to all APIs except DDE for Excel and requires TWS server version 57 or higher.

## Changes to Bond Contract Data

Beginning with TWS Version 921, some bond contract data will be suppressed and will not be available from the API. All bond contract data will continue to be available from Trader Workstation, but only the following bond contract data will be available from the API:

- » Contract ID
- » Minimum Tick
- » CUSIP (if you have subscribed to the CUSIP service)
- » Rating (if you have subscribed to ratings)

## Echo Order Reference Field on Execution Objects

All APIs now echo the order reference field on execution objects. The string attribute orderRef has been added to the Execution object in the Java, Active X and C++ APIs (the order reference field was already available in the DDE for Excel API).

© 2001 - 2012 Interactive Brokers LLC. All rights reserved. Sun, Sun Microsystems, the Sun Logo and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Excel is a trademark or registered trademark of Microsoft Corporation in the United States and/or other countries.