Basic Orders

Auction

An Auction order is entered into the electronic trading system during the pre-market opening period for execution at the Calculated Opening Price (COP). If your order is not filled on the open, the order is re-submitted as a limit order with the limit price set to the COP or the best bid/ask after the market opens.

- Products: FUT, STK
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.Tif = "AUC";
            order.OrderType = "MTL";
            order.TotalQuantity = quantity;
            order.LmtPrice = price;
```

Discretionary

An Discretionary order is a limit order submitted with a hidden, specified 'discretionary' amount off the limit price which may be used to increase the price range over which the limit order is eligible to execute. The market sees only the limit price.

- Products: STK
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = price;
            order.DiscretionaryAmt = discretionaryAmount;
```

Market

A Market order is an order to buy or sell at the market bid or offer price. A market order may increase the likelihood of a fill and the speed of execution, but unlike the Limit order a Market order provides no price protection and may fill at a price far lower/higher than the current displayed bid/ask.

- Products: BOND, CFD, EFP, CASH, FUND, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MKT";
            order.TotalQuantity = quantity;
```

Market If Touched

A Market If Touched (MIT) is an order to buy (or sell) a contract below (or above) the market. Its purpose is to take advantage of sudden or unexpected changes in share or other prices and provides investors with a trigger price to

set an order in motion. Investors may be waiting for excessive strength (or weakness) to cease, which might be represented by a specific price point. MIT orders can be used to determine whether or not to enter the market once a specific price level has been achieved. This order is held in the system until the trigger price is touched, and is then submitted as a market order. An MIT order is similar to a stop order, except that an MIT sell order is placed above the current market price, and a stop sell order is placed below

- Products: BOND, CFD, CASH, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MIT";
            order.TotalQuantity = quantity;
            order.AuxPrice = price;
```

Market On Close

A Market On Close (MOC) order is a market order that is submitted to execute as close to the closing price as possible.

- Products: CFD, FUT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MOC";
            order.TotalQuantity = quantity;
```

Market On Open

A Market On Open (MOO) combines a market order with the OPG time in force to create an order that is automatically submitted at the market's open and fills at the market price.

- Products: CFD, FUT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MKT";
            order.TotalQuantity = quantity;
            order.Tif = "OPG";
```

Pegged to Market

A pegged-to-market order is designed to maintain a purchase price relative to the national best offer (NBO) or a sale price relative to the national best bid (NBB). Depending on the width of the quote, this order may be passive or aggressive. The trader creates the order by entering a limit price which defines the worst limit price that they are willing to accept. Next, the trader enters an offset amount which computes the active limit price as follows: Sell order price = Bid price + offset amount Buy order price = Ask price - offset amount

- Products: STK
- Supported exchanges

```
            Order order = new Order();
```

```
            order.Action = action;
            order.OrderType = "PEG MKT";
            order.TotalQuantity = 100;
            order.AuxPrice = marketOffset;//Offset price
```

Pegged to Stock

A Pegged to Stock order continually adjusts the option order price by the product of a signed user-define delta and the change of the option's underlying stock price. The delta is entered as an absolute and assumed to be positive for calls and negative for puts. A buy or sell call order price is determined by adding the delta times a change in an underlying stock price to a specified starting price for the call. To determine the change in price, the stock reference price is subtracted from the current NBBO midpoint. The Stock Reference Price can be defined by the user, or defaults to the NBBO midpoint at the time of the order if no reference price is entered. You may also enter a high/low stock price range which cancels the order when reached. The delta times the change in stock price will be rounded to the nearest penny in favor of the order.

- Products: OPT
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "PEG STK";
            order.TotalQuantity = quantity;
            order.Delta = delta;
            order.StockRefPrice = stockReferencePrice;
            order.StartingPrice = startingPrice;
```

Pegged to Primary

Relative (a.k.a. Pegged-to-Primary) orders provide a means for traders to seek a more aggressive price than the National Best Bid and Offer (NBBO). By acting as liquidity providers, and placing more aggressive bids and offers than the current best bids and offers, traders increase their odds of filling their order. Quotes are automatically adjusted as the markets move, to remain aggressive. For a buy order, your bid is pegged to the NBB by a more aggressive offset, and if the NBB moves up, your bid will also move up. If the NBB moves down, there will be no adjustment because your bid will become even more aggressive and execute. For sales, your offer is pegged to the NBO by a more aggressive offset, and if the NBO moves down, your offer will also move down. If the NBO moves up, there will be no adjustment because your offer will become more aggressive and execute. In addition to the offset, you can define an absolute cap, which works like a limit price, and will prevent your order from being executed above or below a specified level. Stocks, Options and Futures - not available on paper trading

- Products: CFD, STK, OPT, FUT
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "REL";
            order.TotalQuantity = quantity;
            order.LmtPrice = priceCap;
            order.AuxPrice = offsetAmount;
```

Sweep to Fill

Sweep-to-fill orders are useful when a trader values speed of execution over price. A sweep-to-fill order identifies the best price and the exact quantity offered/available at that price, and transmits the corresponding portion of your order for immediate execution. Simultaneously it identifies the next best price and quantity offered/available, and submits the matching quantity of your order for immediate execution.

- Products: CFD, STK, WAR (SMART only)

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = price;
            order.SweepToFill = true;
```

Auction Limit

For option orders routed to the Boston Options Exchange (BOX) you may elect to participate in the BOX's price improvement auction in pennies. All BOX-directed price improvement orders are immediately sent from Interactive Brokers to the BOX order book, and when the terms allow, IB will evaluate it for inclusion in a price improvement auction based on price and volume priority. In the auction, your order will have priority over broker-dealer price improvement orders at the same price. An Auction Limit order at a specified price. Use of a limit order ensures that you will not receive an execution at a price less favorable than the limit price. Enter limit orders in penny increments with your auction improvement amount computed as the difference between your limit order price and the nearest listed increment.

- Products: OPT (BOX only)

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = price;
            order.AuctionStrategy = auctionStrategy;
```

Auction Pegged to Stock

For option orders routed to the Boston Options Exchange (BOX) you may elect to participate in the BOX's price improvement auction in pennies. All BOX-directed price improvement orders are immediately sent from Interactive Brokers to the BOX order book, and when the terms allow, IB will evaluate it for inclusion in a price improvement auction based on price and volume priority. In the auction, your order will have priority over broker-dealer price improvement orders at the same price. An Auction Pegged to Stock order adjusts the order price by the product of a signed delta (which is entered as an absolute and assumed to be positive for calls, negative for puts) and the change of the option's underlying stock price. A buy or sell call order price is determined by adding the delta times a change in an underlying stock price change to a specified starting price for the call. To determine the change in price, a stock reference price (NBBO midpoint at the time of the order is assumed if no reference price is entered) is subtracted from the current NBBO midpoint. A stock range may also be entered that cancels an order when reached. The delta times the change in stock price will be rounded to the nearest penny in favor of the order and will be used as your auction improvement amount.

- Products: OPT (BOX only)

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "PEG STK";
            order.TotalQuantity = quantity;
            order.Delta = delta;
            order.StartingPrice = startingPrice;
```

Auction Relative

For option orders routed to the Boston Options Exchange (BOX) you may elect to participate in the BOX's price improvement auction in pennies. All BOX-directed price improvement orders are immediately sent from Interactive Brokers to the BOX order book, and when the terms allow, IB will evaluate it for inclusion in a price improvement auction based on price and volume priority. In the auction, your order will have priority over broker-dealer price improvement orders at the same price. An Auction Relative order that adjusts the order price by the product of a signed delta (which is entered as an absolute and assumed to be positive for calls, negative for puts) and the change of the option's underlying stock price. A buy or sell call order price is determined by adding the delta times a change in an underlying stock price change to a specified starting price for the call. To determine the change in price, a stock reference price (NBBO midpoint at the time of the order is assumed if no reference price is entered) is subtracted from the current NBBO midpoint. A stock range may also be entered that cancels an order when reached. The delta times the change in stock price will be rounded to the nearest penny in favor of the order and will be used as your auction improvement amount.

- Products: OPT (BOX only)

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "REL";
            order.TotalQuantity = quantity;
            order.AuxPrice = offset;
```

Block

The Block attribute is used for large volume option orders on ISE that consist of at least 50 contracts. To execute large-volume orders over time without moving the market, use the **Accumulate/Distribute** algorithm.

- Products: OPT
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;//Large volumes!
            order.LmtPrice = price;
            order.BlockOrder = true;
```

Box Top

A Box Top order executes as a market order at the current best price. If the order is only partially filled, the remainder is submitted as a limit order with the limit price equal to the price at which the filled portion of the order executed.

- Products: OPT (BOX only)

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "BOX TOP";
            order.TotalQuantity = quantity;
```

Limit Order

A Limit order is an order to buy or sell at a specified price or better. The Limit order ensures that if the order fills, it will not fill at a price less favorable than your limit price, but it does not guarantee a fill.

- Products: BOND, CFD, CASH, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
```

Forex Cash Quantity Order

Forex orders can be placed in denomination of second currency in pair using cashQty field.
Requires TWS or IBG 963+

- Products: CASH
- Forex Cash Quantity Orders

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.LmtPrice = limitPrice;
            order.CashQty = cashQty;
```

Limit if Touched

A Limit if Touched is an order to buy (or sell) a contract at a specified price or better, below (or above) the market. This order is held in the system until the trigger price is touched. An LIT order is similar to a stop limit order, except that an LIT sell order is placed above the current market price, and a stop limit sell order is placed below.

- Products: BOND, CFD, CASH, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LIT";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
            order.AuxPrice = triggerPrice;
```

Limit on Close

A Limit-on-close (LOC) order will be submitted at the close and will execute if the closing price is at or better than the submitted limit price.

- Products: CFD, FUT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LOC";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
```

Limit on Open

A Limit-on-Open (LOO) order combines a limit order with the OPG time in force to create an order that is submitted at the market's open, and that will only execute at the specified limit price or better. Orders are filled in accordance with specific exchange rules.

- Products: CFD, STK, OPT, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.Tif = "OPG";
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
```

Passive Relative

Passive Relative orders provide a means for traders to seek a less aggressive price than the National Best Bid and Offer (NBBO) while keeping the order pegged to the best bid (for a buy) or ask (for a sell). The order price is automatically adjusted as the markets move to keep the order less aggressive. For a buy order, your order price is pegged to the NBB by a less aggressive offset, and if the NBB moves up, your bid will also move up. If the NBB moves down, there will be no adjustment because your bid will become aggressive and execute. For a sell order, your price is pegged to the NBO by a less aggressive offset, and if the NBO moves down, your offer will also move down. If the NBO moves up, there will be no adjustment because your offer will become aggressive and execute. In addition to the offset, you can define an absolute cap, which works like a limit price, and will prevent your order from being executed above or below a specified level. The Passive Relative order is similar to the Relative/Pegged-to-Primary order, except that the Passive relative subtracts the offset from the bid and the Relative adds the offset to the bid.

- Products: STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "PASSV REL";
            order.TotalQuantity = quantity;
            order.AuxPrice = offset;
```

Pegged to Midpoint

A pegged-to-midpoint order provides a means for traders to seek a price at the midpoint of the National Best Bid and Offer (NBBO). The price automatically adjusts to peg the midpoint as the markets move, to remain aggressive. For a buy order, your bid is pegged to the NBBO midpoint and the order price adjusts automatically to continue to peg the midpoint if the market moves. The price only adjusts to be more aggressive. If the market moves in the opposite direction, the order will execute.

- Products: STK
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "PEG MID";
            order.TotalQuantity = quantity;
            order.AuxPrice = offset;
            order.LmtPrice = limitPrice;
```

Market to Limit

A Market-to-Limit (MTL) order is submitted as a market order to execute at the current best market price. If the order is only partially filled, the remainder of the order is canceled and re-submitted as a limit order with the limit price equal to the price at which the filled portion of the order executed.

- Products: CFD, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MTL";
            order.TotalQuantity = quantity;
```

Market with Protection

This order type is useful for futures traders using Globex. A Market with Protection order is a market order that will be cancelled and resubmitted as a limit order if the entire order does not immediately execute at the market price. The limit price is set by Globex to be close to the current market price, slightly higher for a sell order and lower for a buy order.

- Products: FUT, FOP
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MKT PRT";
            order.TotalQuantity = quantity;
```

Stop

A Stop order is an instruction to submit a buy or sell market order if and when the user-specified stop trigger price is attained or penetrated. A Stop order is not guaranteed a specific execution price and may execute significantly away from its stop price. A Sell Stop order is always placed below the current market price and is typically used to limit a loss or protect a profit on a long stock position. A Buy Stop order is always placed above the current market price. It is typically used to limit a loss or help protect a profit on a short sale.

- Products: CFD, BAG, CASH, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "STP";
            order.AuxPrice = stopPrice;
            order.TotalQuantity = quantity;
```

Stop Limit

A Stop-Limit order is an instruction to submit a buy or sell limit order when the user-specified stop trigger price is attained or penetrated. The order has two basic components: the stop price and the limit price. When a trade has occurred at or through the stop price, the order becomes executable and enters the market as a limit order, which is an order to buy or sell at a specified price or better.

- Products: CFD, CASH, FUT, FOP, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "STP LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
            order.AuxPrice = stopPrice;
```

Stop with Protection

A Stop with Protection order combines the functionality of a stop limit order with a market with protection order. The order is set to trigger at a specified stop price. When the stop price is penetrated, the order is triggered as a market with protection order, which means that it will fill within a specified protected price range equal to the trigger price +/- the exchange-defined protection point range. Any portion of the order that does not fill within this protected range is submitted as a limit order at the exchange-defined trigger price +/- the protection points.

- Products: FUT
- Supported exchanges

```
            Order order = new Order();
            order.TotalQuantity = quantity;
            order.Action = action;
            order.OrderType = "STP PRT";
            order.AuxPrice = stopPrice;
```

Trailing Stop

A sell trailing stop order sets the stop price at a fixed amount below the market price with an attached "trailing" amount. As the market price rises, the stop price rises by the trail amount, but if the stock price falls, the stop loss price doesn't change, and a market order is submitted when the stop price is hit. This technique is designed to allow an investor to specify a limit on the maximum possible loss, without setting a limit on the maximum possible gain. "Buy" trailing stop orders are the mirror image of sell trailing stop orders, and are most appropriate for use in falling markets.

Note that Trailing Stop orders can have the trailing amount specified as a percent, as in the example below, or as an absolute amount which is specified in the auxPrice field.

- Products: CFD, CASH, FOP, FUT, OPT, STK, WAR
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "TRAIL";
            order.TotalQuantity = quantity;
            order.TrailingPercent = trailingPercent;
```

```
            order.TrailStopPrice = trailStopPrice;
```

Trailing Stop Limit

A trailing stop limit order is designed to allow an investor to specify a limit on the maximum possible loss, without setting a limit on the maximum possible gain. A SELL trailing stop limit moves with the market price, and continually recalculates the stop trigger price at a fixed amount below the market price, based on the user-defined "trailing" amount. The limit order price is also continually recalculated based on the limit offset. As the market price rises, both the stop price and the limit price rise by the trail amount and limit offset respectively, but if the stock price falls, the stop price remains unchanged, and when the stop price is hit a limit order is submitted at the last calculated limit price. A "Buy" trailing stop limit order is the mirror image of a sell trailing stop limit, and is generally used in falling markets.

Trailing Stop Limit orders can be sent with the trailing amount specified as an absolute amount, as in the example below, or as a percentage, specified in the trailingPercent field.

Important: the 'limit offset' field is set by default in the TWS/IBG settings in v963+. This setting either needs to be changed in the Order Presets, the default value accepted, or the limit price offset sent from the API as in the example below. Not both the 'limit price' and 'limit price offset' fields can be set in TRAIL LIMIT orders.

- Products: BOND, CFD, CASH, FUT, FOP, OPT, STK, WAR

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "TRAIL LIMIT";
            order.TotalQuantity = quantity;
            order.TrailStopPrice = trailStopPrice;
            order.LmtPriceOffset = lmtPriceOffset;
            order.AuxPrice = trailingAmount;
```

Combo Limit

Create combination orders that include options, stock and futures legs (stock legs can be included if the order is routed through SmartRouting). Although a combination/spread order is constructed of separate legs, it is executed as a single transaction if it is routed directly to an exchange. For combination orders that are SmartRouted, each leg may be executed separately to ensure best execution.

- Products: OPT, STK, FUT

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.LmtPrice = limitPrice;
            if (nonGuaranteed)
            {
                order.SmartComboRoutingParams = new List<TagValue>();
                order.SmartComboRoutingParams.Add(new TagValue("NonGuaranteed", "1"));
            }
```

Combo Market

Create combination orders that include options, stock and futures legs (stock legs can be included if the order is routed through SmartRouting). Although a combination/spread order is constructed of separate legs, it is executed

as a single transaction if it is routed directly to an exchange. For combination orders that are SmartRouted, each leg may be executed separately to ensure best execution.

- Products: OPT, STK, FUT

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "MKT";
            order.TotalQuantity = quantity;
            if (nonGuaranteed)
            {
                order.SmartComboRoutingParams = new List<TagValue>();
                order.SmartComboRoutingParams.Add(new TagValue("NonGuaranteed", "1"));
            }
```

Combo Limit with Price per Leg

Create combination orders that include options, stock and futures legs (stock legs can be included if the order is routed through SmartRouting). Although a combination/spread order is constructed of separate legs, it is executed as a single transaction if it is routed directly to an exchange. For combination orders that are SmartRouted, each leg may be executed separately to ensure best execution.

- Products: OPT, STK, FUT

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "LMT";
            order.TotalQuantity = quantity;
            order.OrderComboLegs = new List<OrderComboLeg>();
            foreach(double price in legPrices)
            {
                OrderComboLeg comboLeg = new OrderComboLeg();
                comboLeg.Price = 5.0;
                order.OrderComboLegs.Add(comboLeg);
            }
            if (nonGuaranteed)
            {
                order.SmartComboRoutingParams = new List<TagValue>();
                order.SmartComboRoutingParams.Add(new TagValue("NonGuaranteed", "1"));
            }
```

Relative Limit Combo

Create combination orders that include options, stock and futures legs (stock legs can be included if the order is routed through SmartRouting). Although a combination/spread order is constructed of separate legs, it is executed as a single transaction if it is routed directly to an exchange. For combination orders that are SmartRouted, each leg may be executed separately to ensure best execution.

- Products: OPT, STK, FUT

```
            Order order = new Order();
            order.Action = action;
            order.TotalQuantity = quantity;
            order.OrderType = "REL + LMT";
            order.LmtPrice = limitPrice;
            if (nonGuaranteed)
            {
                order.SmartComboRoutingParams = new List<TagValue>();
                order.SmartComboRoutingParams.Add(new TagValue("NonGuaranteed", "1"));
            }
```

Relative Market Combo

Create combination orders that include options, stock and futures legs (stock legs can be included if the order is routed through SmartRouting). Although a combination/spread order is constructed of separate legs, it is executed as a single transaction if it is routed directly to an exchange. For combination orders that are SmartRouted, each leg may be executed separately to ensure best execution.

- Products: OPT, STK, FUT

```
            Order order = new Order();
            order.Action = action;
            order.TotalQuantity = quantity;
            order.OrderType = "REL + MKT";
            if (nonGuaranteed)
            {
                order.SmartComboRoutingParams = new List<TagValue>();
                order.SmartComboRoutingParams.Add(new TagValue("NonGuaranteed", "1"));
            }
```

Volatility

Specific to US options, investors are able to create and enter Volatility-type orders for options and combinations rather than price orders. Option traders may wish to trade and position for movements in the price of the option determined by its implied volatility. Because implied volatility is a key determinant of the premium on an option, traders position in specific contract months in an effort to take advantage of perceived changes in implied volatility arising before, during or after earnings or when company specific or broad market volatility is predicted to change. In order to create a Volatility order, clients must first create a Volatility Trader page from the Trading Tools menu and as they enter option contracts, premiums will display in percentage terms rather than premium. The buy/sell process is the same as for regular orders priced in premium terms except that the client can limit the volatility level they are willing to pay or receive.

- Products: FOP, OPT
- Supported exchanges

```
            Order order = new Order();
            order.Action = action;
            order.OrderType = "VOL";
            order.TotalQuantity = quantity;
            order.Volatility = volatilityPercent;//Expressed in percentage (40%)
            order.VolatilityType = volatilityType;// 1=daily, 2=annual
```

Pegged to Benchmark

The Pegged to Benchmark order is similar to the Pegged to Stock order for options, except that the Pegged to Benchmark allows you to specify any asset type as the reference (benchmark) contract for a stock or option order. Both the primary and reference contracts must use the same currency.

- Products: STK, OPT
- Supported exchanges

```
            Order order = new Order();
            order.OrderType = "PEG BENCH";
            //BUY or SELL
            order.Action = action;
            order.TotalQuantity = quantity;
```

```
            //Beginning with price...
            order.StartingPrice = startingPrice;
            //increase/decrease price..
            order.IsPeggedChangeAmountDecrease = peggedChangeAmountDecrease;
            //by... (and likewise for price moving in opposite direction)
            order.PeggedChangeAmount = peggedChangeAmount;
            //whenever there is a price change of...
            order.ReferenceChangeAmount = referenceChangeAmount;
            //in the reference contract...
            order.ReferenceContractId = referenceConId;
            //being traded at...
            order.ReferenceExchange = referenceExchange;
            //starting reference price is...
            order.StockRefPrice = stockReferencePrice;
            //Keep order active as long as reference contract trades between...
            order.StockRangeLower = referenceContractLowerRange;
            //and...
            order.StockRangeUpper = referenceContractUpperRange;
```

Limit Order With Manual Order Time

The Limit Order With Manual Order Time is a **Limit Order** with ManualOrderTime property.

```
            Order order = OrderSamples.LimitOrder(action, quantity, limitPrice);
            order.ManualOrderTime = manualOrderTime;
```

Placing a Limit Order With Manual Order Time

The *Manual Order Time* field is required for, and becomes editable for, "manual" orders, which are orders that originate from a client (whether by phone, email, chat, verbally from the floor, from another desk, etc.) and are then entered into the system.

The *Manual Order Time* field is not used for client orders received electronically, nor for orders that originate from the account manager (with discretion) in the client's accounts, or when an order is allocated to ALL accounts or among multiple accounts using an account group or model portfolio.

```
            client.placeOrder(nextOrderId++, ContractSamples.USStockAtSmart(),
        OrderSamples.LimitOrderWithManualOrderTime("BUY", Util.StringToDecimal("100"), 111.11,
        "20220314-13:00:00"));
```

Canceling a Limit Order With Manual Order Time

The *Manual Order Cancel Time* field is required for, and becomes editable for, "manual" order cancelations, which are order cancelations that originate from a client (whether by phone, email, chat, verbally from the floor, from another desk, etc.) and are then entered into the system.

The *Manual Order Cancel Time* field is not used for client orders cancelations received electronically, nor for orders that originate from the account manager (with discretion) in the client's accounts, or when an order is allocated to ALL accounts or among multiple accounts using an account group or model portfolio.

```
            client.cancelOrder(nextOrderId - 1, "20220314-19:00:00");
```