Documentation for the new reserved words in MultiCharts 7

www.MultiCharts.com

Contents of this document

Source:
forum post, MultiCharts Blog post.
Document version: 10/7/2011.

The PosTrade.. reserved words for Signals

1. PosTradeCount

Returns a numerical value, indicating the total number of entries for the specified position.

Usage:

```
PosTradeCount (PosBack)
```

Where:

```
PosBack - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
```

Notes

This function can only be used in signals.

Example

PosTradeCount (1) will return a value of 2 if there were two separate entries for the most recently closed position.

2. PosTradeSize

Returns an absolute numerical value, indicating the number of contracts or shares in the specified trade.

Usage:

```
PosTradeSize(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeSize(0,1) will return a value of 2 for the second trade of the open position, if this trade had a quantity of 2.

3. PosTradeCommission

Returns an absolute numerical value, indicating the commission amount spent for the specified trade.

Usage:

```
PosTradeCommission(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeCommission(0,1) will return a value of 5 for the second trade of the open position, if the commission for this trade is 5 dollars.

4. PosTradeProfit

Returns an absolute numerical value, indicating the profit (or loss if negative) of the specified trade.

Usage:

```
PosTradeProfit(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeProfit(0,1) will return a value of 3 for the second trade of the open position, if the profit for this trade is 3 dollars.

5. PosTradeEntryName

Returns entry order name. Entry Name is indicated on the chart and in Order and Position Tracker Window.

Usage:

```
PosTradeEntryName(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeEntryName(0,1) will return a value of “buy LE” for the second trade of the open position, if this trade was opened by the order with “buy LE” name.

6. PosTradeEntryPrice

Returns an absolute numerical value, indicating the execution price of trade entry order.

Usage:

```
PosTradeEntryPrice(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
0 - open position;
1 - one position back (the last position closed);
2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeEntryPrice(0,1) will return a value of 100.2 for the second trade of the open position, if this trade was opened by the order filled at 100.2.

7. PosTradeEntryBar

Returns an absolute numerical value, indicating bar number of the trade entry order.

Usage

```
PosTradeEntryBar(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeEntryBar(0,1) will return a value of 25 for the second trade of the open position, if this trade was opened on 25th bar.

8. PosTradeEntryCategory

Returns an absolute numerical value, indicating trade entry order category. The following types are possible:

1 = Stop order (buy next bar at close - 1 point stop);

2 = Limit order (buy next bar at close + 1 point limit);

3 = Market order (buy next bar market);

4 = Market at Close order (buy this bar at close);

5 = Market at open order (buy next bar open);

8 = StopLimit order (buy 1 contracts next bar at close - 2 point stop close + 2 point limit).

Usage

```
PosTradeEntryCateory(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeEntryCategory(0,1) will return a value of 1 for the second trade of the open position, if the order type was Stop Order.

9. PosTradeExitName

Returns exit order name. Exit Name is indicated on the chart and in Order and Position Tracker Window.

Usage

```
PosTradeExitName(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount

Notes

This function can only be used in signals.

Example

PosTradeExitName(0,1) will return a value of “sell LX” for the second trade of the open position, if this trade was closed by the order with “sell LX” name.

10. PosTradeExitPrice

Returns an absolute numerical value, indicating the execution price of trade exit order.

Usage:

```
PosTradeExitPrice(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeExitPrice(0,1) will return a value of 100.5 for the second trade of the open position, if this trade was closed by the order filled at 100.5.

11. PosTradeExitBar

Returns an absolute numerical value, indicating bar number of the trade exit order.

Usage:

```
PosTradeExitBar(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeExitBar(0,1) will return a value of 28 for the second trade of the open position, if this trade was closed on 28th bar.

12. PosTradeExitCategory

Returns an absolute numerical value, indicating trade exit order category. The following types are possible:

1 = Stop order (buy next bar at close - 1 point stop)

2 = Limit order (buy next bar at close + 1 point limit)

3 = Market order (buy next bar market)

4 = Market at Close order (buy this bar at close)

5 = Market at open order (buy next bar open)

8 = StopLimit order (buy 1 contracts next bar at close - 2 point stop close + 2 point limit)

Usage

```
PosTradeExitCateory(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Example

PosTradeExitCategory(0,1) will return a value of 3 for the second trade of the open position, if the closing order type was Market Order.

13. PosTradeIsOpen

Returns True value if the trade is open, False value if the trade is closed. It makes sense to check the trades of the open position. For other positions False is always returned.

Usage:

```
PosTradeIsOpen(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount

Notes

This function can only be used in signals.

Example

PosTradeIsOpen(0,1) will return True for the second trade of the open position, if this trade is opened (haven't close order).

14. PosTradeIsLong

Returns True value if the trade was opened by buy order, otherwise False value is returned.

Usage:

```
PosTradeIsLong(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount

Notes

This function can only be used in signals.

Example

PosTradeIsLong(0,1) will return True for the second trade of the open position, if this trade was opened by "buy" order.

15. PosTradeEntryDateTime

Returns double-precision decimal DateTime for entry order. As an example see **computerdatetime**.

Usage

```
PosTradeEntryDateTime(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

16. PosTradeExitDateTime

Returns double-precision decimal DateTime for exit order. As an example see **computerdatetime**.

Usage:

```
PosTradeExitDateTime(PosAgo, TradeNumber)
```

Where:

```
PosAgo - a numerical expression, specifying the position:
      0 - open position;
      1 - one position back (the last position closed);
      2 - two positions back, etc.
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in specified position use PosTradeCount.

Notes

This function can only be used in signals.

Portfolio reserved words

17. Portfolio_InvestedCapital

Returns absolute value in US dollars, indicating the amount of cash assets invested in portfolio securities on the moment of strategy calculation.

Informational reserved words

18. Entryname

Returns the name of the order which opened the position.

Usage

```
EntryName(TradeNumber)
```

Where:

```
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

Notes

Same as PosTradeEntryName(0, TradeNumber).

19. Exitname

Returns the name of the order which closed the position.

Usage:

```
ExitName(TradeNumber)
```

Where:

```
TradeNumber - a numerical expression, specifying the number of trade
(zero-based).
```

Notes

Same as PosTradeExitName(0, TradeNumber).

20. AvgEntryPrice_at_Broker_for_The_Strategy

Returns a numerical value, indicating the average entry price at the broker for the strategy.

A positive value indicates a long position and a negative value indicates a short position. A zero ('0') is returned when the current position is flat, or if Automated Trading is not turned on.

Usage

```
AvgEntryPrice_at_Broker_for_The_Strategy
```

Notes

This function can only be used in signals and functions.

Important

If Automated Trading was manually turned off by the user, the value returned by the keyword stops changing, and may remain unequal to '0'.

21. AvgEntryPrice_at_Broker

Returns a numerical value, indicating the average entry price at the broker for the symbol.

A positive value indicates a long position and a negative value indicates a short position.

A zero ('0') is returned when the current position is flat, or if Automated Trading is not turned on.

Usage:

```
AvgEntryPrice_at_Broker
```

Notes

This function can only be used in signals and functions.

This function can only be used with Interactive Brokers, Patsystems, and Zen-Fire.

Important

If Automated Trading was manually turned off by the user, the value returned by the keyword stops changing, and may remain unequal to '0'.

22. q_time_s

Same as **q_time**. Time is indicated in HHmmss format.

23. ClearPrintLog

Same as **cleardebug**.

24. mc_tl_getactive

Returns a numerical value indicating the trendline ID number of the currently selected trendline; returns a value of -1 if no trendlines are currently selected.

Usage

```
Mc_tl_GetActive
```

Notes

A trendline-specific ID number is assigned by mc_tl_New when the trendline is created.

Example

Assign a value, indicating the trendline ID number of the currently selected trendline, to Value1 variable:

```
Value1=mc_TL_GetActive;
```

25. tl_getactive

This reserved word returns a numeric value representing the ID of the currently active trendline..

```
Value1 = TL_GetActive();
```

Value1 is any numeric variable or array. You must assign the trendline reserved word to a numeric variable or array so that you can determine whether or not the reserved word performed its operation successfully.

Remarks

When the reserved word performs its operation successfully, a 0 is returned. When a reserved word cannot perform its operation, it returns an error code.

26. mc_arw_getactive

Returns a numerical value indicating the arrow ID number of the currently selected arrow; returns a value of -1 if no arrows are currently selected.

Usage

```
Mc_arw_GetActive
```

Example

Assign a value, indicating the arrow ID number of the currently selected arrow, to Value1 variable:

```
Value1=mc_arw_GetActive;
```

27. mc_text_getactive

Returns a numerical value indicating the text ID number of the currently selected text; returns a value of -1 if no text is currently selected.

Usage

```
Mc_text_GetActive
```

Example

Assign a value, indicating the text ID number of the currently selected text, to Value1 variable:

```
Value1=mc_text_GetActive;
```

28. Boxsize

Returns the price-based interval setting associated with the specified price-based chart type an indicator or signal is applied to.

This value is set in the Chart Type section of the Settings tab within the Format Instrument dialog for a chart.

Chart Type	Value Returned
Point & Figure	Box Size value
Point	Point value
Renko	Box Size value

29. Revsize

Returns the Reversal of a Point & Figure chart, the Reversalof a Kagi chart, or the number of Line Breaks in a Line Break chart. This value is set in the Chart Type section of the Settings tab within the Format Instrument dialog for a chart.

Example

RevSize returns 2 if the reversal size of a P & F chart is set to 2.

RevSize returns 5 if the number of line breaks for a Line Break chart is set to 5.

RevSize returns 4 if the reversal size of a Kagi chart is set to 4%.

30. RecalcLastBarAfter

Initializes the calculation after expiration of the timeout, set in seconds

Usage

```
RecalcLastBarAfter(timeout)
```

Where timeout indicates the number of seconds.

Notes

RecalcLastBarAfter is used for the slow markets. Calculation of the studies is performed in events when the immediate tick has come to a chart. When the tick has not come since the moment of the last calculation and during timeout (sec) a new calculation is being initialized. When the tick has been received before timeout expiration, time counter is reset and countdown for RecalcLastBarAfter(timeout) starts anew.

31. i_AvgEntryPrice_at_Broker

Returns the Average entry price of each open entry in a pyramided position.

Notes

I_AvgEntryPrice only returns the average entry price for open trades.

I_AvgEntryPrice can only be used in an indicator.

I_AvgEntryPrice will only return a value if a signal is applied to the same data.

Example

I_AvgEntryPrice returns 170 if three trades are currently open and were entered at a price of 140, 170, and 200.

I_AvgEntryPrice returns 53 if four trades are currently open and were entered at a price of 54, 48, 60, and 50.

32. i_AvgEntryPrice_at_Broker_for_The_Strategy

Is used for the extraction of strategy information in indicator. Returns the same information as AvgEntryPrice_at_Broker_for_The_Strategy.

Reserved words for submitting orders

33. PlaceMarketOrder

Places market order at the broker without position changing on the chart

Usage:

```
PlaceMarketOrder(IsBuy, IsEntry, Contracts)
```

Where:

```
IsBuy indicates whether order is buy or sell,
IsEntry indicates whether order is entry or exit,
Contracts indicates the number of contracts/shares of the order.
```

Notes

Works with auto trading turned off. Can be used as a mean of synchronization of strategy market position with a broker.

34. ChangeMarketPosition

Places the order with set name and price on the chart. Though when auto trading is off the order will not be sent to a broker.

Usage

```
ChangeMarketPosition(Delta, Price, Name)
```

Where

```
Delta – number of contracts by which current market position should
be changed.
Price – order filling price.
Name – name of the order that changes the position.
```

Notes

Can be used as a mean of synchronization of strategy market position with a broker

Example

```
If marketposition = 2 then ChangeMarketPosition(-2, 100, "LX");
```

Will place close order with the name "LX" and the price 100 if current marketposition =2.

```
If marketposition = 0 then ChangeMarketPosition(-2, 100, "SE");
```

Will place open order with the name "SE" and the price 100 if current marketposition =0.

The Open.. reserved words for Signals

35. OpenEntriesCount

Same as **CurrentEntries**.

36. OpenEntryDate

Returns a numerical value, indicating the date of specified entry into the open position. The date is indicated in the YYYMMdd format, where YYY is the number of years since 1900, MM is the month, and dd is the day of the month.

Usage:

```
OpenEntryDate(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount

Notes

This function can only be used in signals.

Example

OpenEntryDate(1) will return 1110402 for the open position if the second trade was generated at April 2nd, 2011.

37. OpenEntryTime

Returns a numerical value, indicating the time of specified entry into the open position. The time is indicated in the HHmm format, where HH is the hour in 24 hours format and mm are minutes.

Usage:

```
OpenEntryTime(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount

Notes

This function can only be used in signals.

38. OpenEntryPrice

Returns a numerical value, indicating the price of specified entry into the open position.

Usage:

```
OpenEntryPrice(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount

Notes

This function can only be used in signals.

39. OpenEntryContracts

Returns a numerical value, indicating the Quantity of contracts of specified entry order into the open position.

Usage

```
OpenEntryContracts(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

40. OpenEntryProfit

Returns a numerical value, indicating the profit (loss if negative) of specified entry into the open position in dollars.

Usage:

```
OpenEntryProfit(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

41. OpenEntryMaxProfit

Returns a numerical value, indicating maximal value of OpenEntryProfit for the time from entry order execution.

Usage:

```
OpenEntryMaxProfit(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

42. OpenEntryMinProfit

Returns a numerical value, indicating minimal value of OpenEntryProfit for the time from entry order execution

Usage:

```
OpenEntryMinProfit(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

43. OpenEntryProfitPerContract

Returns a numerical value, indicating the profit (loss if negative) per contract of specified entry in dollars.

Usage:

```
OpenEntryProfitPerContract(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

44. OpenEntryMaxProfitPerContract

Returns a numerical value, indicating maximal value of OpenEntryProfitPerContract for the time from entry order execution

Usage:

```
OpenEntryMaxProfitPerContract(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

45. OpenEntryMinProfitPerContract

Returns a numerical value, indicating minimal value of **OpenEntryProfitPerContract** for the time from entry order execution.

Usage:

```
OpenEntryMinProfitPerContract(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

46. OpenEntryComission

Returns a numerical value, indicating the amount of cash assets in US dollars spent on the commission for specified trade.

Usage

```
OpenEntryComission(EntryIndex)
```

Where:

```
EntryIndex - a numerical expression, specifying the number of trade
(zero-based).
```

To retrieve the total number of trades in open position use OpenEntriesCount.

Notes

This function can only be used in signals.

Other new reserved words

47. Not

Used in TrueFalse statements – negative

Example

```
condition1 = true;
condition2 = not condition1;
```

Assigns to condition2 value opposite to condition1.

Other changes in the MultiCharts 7 reserved words

i_openequity

Returns the current equity = netprofit + openpositionprofit

Usage

```
i_OpenEquity
```

Notes

This function can only be used in studies.

Example

i_OpenEquity will return 100 if the current equity is 100.

i_OpenEquity will return -100 if the current equity is -100.