OrderSend

The main function used to open market or place a pending order.

```
int  OrderSend(
   string   symbol,              // symbol
   int      cmd,                 // operation
   double   volume,              // volume
   double   price,               // price
   int      slippage,            // slippage
   double   stoploss,            // stop loss
   double   takeprofit,          // take profit
   string   comment=NULL,        // comment
   int      magic=0,             // magic number
   datetime expiration=0,        // pending order expiration
   color    arrow_color=clrNONE  // color
   );
```

Parameters

symbol
[in] Symbol for trading.

cmd
[in] Operation type. It can be any of the Trade operation enumeration.

volume
[in] Number of lots.

price
[in] Order price.

slippage
[in] Maximum price slippage for buy or sell orders.

stoploss
[in] Stop loss level.

takeprofit
[in] Take profit level.

comment=NULL
[in] Order comment text. Last part of the comment may be changed by server.

magic=0
[in] Order magic number. May be used as user defined identifier.

expiration=0
[in] Order expiration time (for pending orders only).

arrow_color=clrNONE
[in] Color of the opening arrow on the chart. If parameter is missing or has CLR_NONE value opening arrow is not drawn on the chart.

Returned value

Returns number of the ticket assigned to the order by the trade server or -1 if it fails. To get additional error information, one has to call the GetLastError() function.

Note

At opening of a market order (OP_SELL or OP_BUY), only the latest prices of Bid (for selling) or Ask (for buying) can be used as open price. If operation is performed with a security differing from the current one, the MarketInfo() function must be used with MODE_BID or MODE_ASK parameter for the latest quotes for this security to be obtained.

Calculated or unnormalized price cannot be applied. If there has not been the requested open price in the price thread or it has not been normalized according to the amount of digits after decimal point, the error 129 (ERR_INVALID_PRICE) will be generated. If the requested open price is fully out of date, the error 138 (ERR_REQUOTE) will be generated independently on the slippage parameter. If the requested price is out of date, but present in the thread, the order will be opened at the current price and only if the current price lies within the range of price+-slippage.

StopLoss and TakeProfit levels cannot be too close to the market. The minimal distance of stop levels in points can be obtained using the MarketInfo() function with MODE_STOPLEVEL parameter. In the case of erroneous or unnormalized stop levels, the error 130 (ERR_INVALID_STOPS) will be generated. A zero value of MODE_STOPLEVEL means either absence of any restrictions on the minimal distance for Stop Loss/Take Profit or the fact that a trade server utilizes some external mechanisms for dynamic level control, which cannot be translated in the client

terminal. In the second case, GetLastError() can return error 130, because MODE_STOPLEVEL is actually "floating" here.

At placing of a pending order, the open price cannot be too close to the market. The minimal distance of the pending price from the current market one in points can be obtained using the MarketInfo() function with the MODE_STOPLEVEL parameter. In case of false open price of a pending order, the error 130 (ERR_INVALID_STOPS) will be generated.

Applying of pending order expiration time can be disabled in some trade servers. In this case, when a non-zero value is specified in the expiration parameter, the error 147 (ERR_TRADE_EXPIRATION_DENIED) will be generated.

On some trade servers, the total amount of open and pending orders can be limited. If this limit has been exceeded, no new order will be opened (or no pending order will be placed) and trade server will return error 148 (ERR_TRADE_TOO_MANY_ORDERS).

Example:

```
//+------------------------------------------------------------------+
//| Script program start function                                    |
//+------------------------------------------------------------------+
void OnStart()
  {
//--- get minimum stop level
   double minstoplevel=MarketInfo(Symbol(),MODE_STOPLEVEL);
   Print("Minimum Stop Level=",minstoplevel," points");
   double price=Ask;
//--- calculated SL and TP prices must be normalized
   double stoploss=NormalizeDouble(Bid-minstoplevel*Point,Digits);
   double takeprofit=NormalizeDouble(Bid+minstoplevel*Point,Digits);
//--- place market order to buy 1 lot
   int ticket=OrderSend(Symbol(),OP_BUY,1,price,3,stoploss,takeprofit,"My order",16384,0,clrGreen);
   if(ticket<0)
     {
      Print("OrderSend failed with error #",GetLastError());
     }
   else
      Print("OrderSend placed successfully");
//---
  }
```