



OrderSelect

The function selects an order for further processing.

```
bool OrderSelect(
    int    index,           // index or order ticket
    int    select,          // flag
    int    pool=MODE_TRADES // mode
);
```

Parameters

ticket

[in] Order index or order ticket depending on the second parameter.

select

[in] Selecting flags. It can be any of the following values:

SELECT_BY_POS - index in the order pool,
SELECT_BY_TICKET - index is order ticket.

pool=MODE_TRADES

[in] Optional order pool index. Used when the selected parameter is SELECT_BY_POS. It can be any of the following values:

MODE_TRADES (default)- order selected from trading pool(opened and pending orders),
MODE_HISTORY - order selected from history pool (closed and canceled order).

Returned value

It returns true if the function succeeds, otherwise false. To get the [error](#) information, one has to call the [GetLastError\(\)](#) function.

Note

The pool parameter is ignored if the order is selected by the ticket number. The ticket number is a unique order identifier.

To find out from what list the order has been selected, its close time must be analyzed. If the order close time equals to 0, the order is open or pending and taken from the terminal open orders list.

One can distinguish an opened order from a pending order by the order type. If the order close time does not equal to 0, the order is a closed order or a deleted pending order and was selected from the terminal history. They also differ from each other by their order types.

The OrderSelect() function copies order data into program environment and all further calls of [OrderClosePrice\(\)](#), [OrderCloseTime\(\)](#), [OrderComment\(\)](#), [OrderCommission\(\)](#), [OrderExpiration\(\)](#), [OrderLots\(\)](#), [OrderMagicNumber\(\)](#), [OrderOpenPrice\(\)](#), [OrderOpenTime\(\)](#), [OrderPrint\(\)](#), [OrderProfit\(\)](#), [OrderStopLoss\(\)](#), [OrderSwap\(\)](#), [OrderSymbol\(\)](#), [OrderTakeProfit\(\)](#), [OrderTicket\(\)](#), [OrderType\(\)](#) functions return the data, copied earlier. It means that in some cases the order details (open price, SL/TP levels or expiration date) may change and the data become non-actual. It is strongly recommended to call the OrderSelect() function before request the order data.

Consecutive selection of orders using the SELECT_BY_POS parameter returns information in the sequence in which it was received from the trading server. Sorting of the resulting list of orders cannot be guaranteed.

Example:

```
if(OrderSelect(12470, SELECT_BY_TICKET)==true)
{
    Print("order #12470 open price is ", OrderOpenPrice());
    Print("order #12470 close price is ", OrderClosePrice());
}
else
    Print("OrderSelect returned the error of ", GetLastError());
```

See also

[Order properties](#), [OrderClosePrice\(\)](#), [OrderCloseTime\(\)](#), [OrderComment\(\)](#), [OrderCommission\(\)](#), [OrderExpiration\(\)](#), [OrderLots\(\)](#), [OrderMagicNumber\(\)](#), [OrderOpenPrice\(\)](#), [OrderOpenTime\(\)](#), [OrderPrint\(\)](#), [OrderProfit\(\)](#), [OrderStopLoss\(\)](#), [OrderSwap\(\)](#), [OrderSymbol\(\)](#), [OrderTakeProfit\(\)](#), [OrderTicket\(\)](#), [OrderType\(\)](#)