

DDS2 Agent API

For Windows

Programming Guide

Revision 2.0

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Project Scope.....	1
2. Overall Description	1
2.1 Product Features	1
2.2 Environment Schema.....	1
2.3 Operating Environment	1
2.4 Hardware Environment	1
2.5 User Documentation.....	2
3. Functions overview	2
3.1 SubmitOrder	2
3.2 ClosePosition.....	3
3.3 CloseProfitPosition.....	3
3.4 CancelOrder.....	3
3.5 SubmitStop	4
3.6 PlaceOffer.....	4
4. Error codes	5
Appendix A: MQL4	6
Appendix B: EasyLanguage.....	7

Revision History

Date	Reason For Changes	Version
01/11/06	Basic updates.	1.6
02/11/06	BUYLIMIT, BUYSTOP, SELLLIMIT, and SELLSTOP orders are added to the SubmitOrder function	1.7
06/11/06	A new function CloseProfitPosition is added.	1.8
09/11/06	The opposite sides for BUYLIMIT, BUYSTOP, SELL LIMIT, SELLSTOP orders and StopLosses are added to the SubmitOrder function.	1.9
06/12/06	A new function PlaceOffer is added. Strategies executor is added.	2.0

1. Introduction

1.1 Project Scope

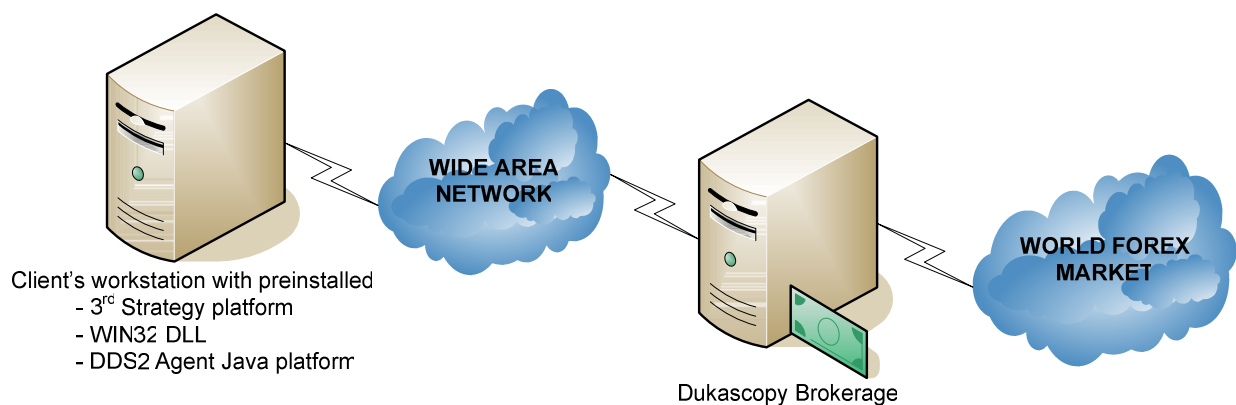
DDS2 Agent API is a new programming interface for automated trading. The following solution is used to transfer trading signals from the 3rd Strategy platform directly to the DDS2 Agent Java trading platform.

2. Overall Description

2.1 Product Features

The trading signals are being initiated by 3rd platform strategy functions. Each strategy function has to be duplicated by DDS2 Agent API function. The DDS2 Agent API functions are called from external WIN32 DLL. The WIN32 DLL is provided by the DDS2 Agent Java platform. All trading transactions between 3rd Strategy platform and WIN32 DLL are captured by DDS2 Agent Java platform, i.e. all trading signals of 3rd Strategy platform are transmitted automatically to DDS2 Agent Java platform.

2.2 Environment Schema



2.3 Operating Environment

- Microsoft Windows XP or Windows 2000
- Sun Java Runtime Environment 5.0+
- MetaTrader 4
- Tradestation
- any 3rd Strategy platform that supports external DLL exporting

2.4 Hardware Environment

- CPU 1500 MHz+

- RAM 512 MB+

2.5 User's Documentation

[DEMO DDS2 Agent Java Platform](#)

[DDS2 Agent API](#)

3. Functions overview

3.1 SubmitOrder

```
int SubmitOrder (string reference, string instruments, int side,
double amount, double price, int stoploss, int takeprofit);
```

The function is used to place a market order or an entry order. The function returns a number of the opened position in DDS2 platform or a number of a position in DDS2 platform that will be opened or [error code](#) in case of failure.

Parameters	Description																																																							
reference	Mandatory. A unique identifier. Maximum 8 characters allowed.																																																							
instruments	Mandatory. Instruments for trading. CCY1/CCY2.																																																							
side	Mandatory. Type of operation. <table><tr><th>Side</th><th>Type</th><th>Description</th><th>SL</th><th>TP</th></tr><tr><td>0</td><td>BUY</td><td>BUY market order.</td><td>BID <</td><td>BID ></td></tr><tr><td>1</td><td>SELL</td><td>SELL market order.</td><td>ASK ></td><td>ASK <</td></tr><tr><td>2</td><td>BUYLIMIT</td><td>Buy Limit entry order. ASK <</td><td>BID <</td><td>BID ></td></tr><tr><td>3</td><td>SELLLIMIT</td><td>Sell Limit entry order. BID ></td><td>ASK ></td><td>ASK <</td></tr><tr><td>4</td><td>BUYSTOP</td><td>Buy Stop entry order. ASK ></td><td>BID <</td><td>BID ></td></tr><tr><td>5</td><td>SELLSTOP</td><td>Sell Stop entry order. BID <</td><td>ASK ></td><td>ASK <</td></tr><tr><td>6</td><td>BUYLIMIT_BY_BID</td><td>Buy Limit entry order. BID <</td><td>BID <</td><td>BID ></td></tr><tr><td>7</td><td>SELLLIMIT_BY_ASK</td><td>Sell Limit entry order. ASK ></td><td>ASK ></td><td>ASK <</td></tr><tr><td>8</td><td>BUYSTOP_BY_BID</td><td>Buy Stop entry order. BID ></td><td>BID <</td><td>BID ></td></tr><tr><td>9</td><td>SELLSTOP_BY_ASK</td><td>Sell Stop entry order. ASK <</td><td>ASK ></td><td>ASK <</td></tr></table> <p>Directions for StopLoss (SL) and TakeProfit (TP) conditions will be set according to a side parameter. To define another direction of SL or TP condition, do not define these conditions in SubmitOrder function, but use SubmitStop function. To delete SL or TP condition, use SubmitStop function.</p>	Side	Type	Description	SL	TP	0	BUY	BUY market order.	BID <	BID >	1	SELL	SELL market order.	ASK >	ASK <	2	BUYLIMIT	Buy Limit entry order. ASK <	BID <	BID >	3	SELLLIMIT	Sell Limit entry order. BID >	ASK >	ASK <	4	BUYSTOP	Buy Stop entry order. ASK >	BID <	BID >	5	SELLSTOP	Sell Stop entry order. BID <	ASK >	ASK <	6	BUYLIMIT_BY_BID	Buy Limit entry order. BID <	BID <	BID >	7	SELLLIMIT_BY_ASK	Sell Limit entry order. ASK >	ASK >	ASK <	8	BUYSTOP_BY_BID	Buy Stop entry order. BID >	BID <	BID >	9	SELLSTOP_BY_ASK	Sell Stop entry order. ASK <	ASK >	ASK <
Side	Type	Description	SL	TP																																																				
0	BUY	BUY market order.	BID <	BID >																																																				
1	SELL	SELL market order.	ASK >	ASK <																																																				
2	BUYLIMIT	Buy Limit entry order. ASK <	BID <	BID >																																																				
3	SELLLIMIT	Sell Limit entry order. BID >	ASK >	ASK <																																																				
4	BUYSTOP	Buy Stop entry order. ASK >	BID <	BID >																																																				
5	SELLSTOP	Sell Stop entry order. BID <	ASK >	ASK <																																																				
6	BUYLIMIT_BY_BID	Buy Limit entry order. BID <	BID <	BID >																																																				
7	SELLLIMIT_BY_ASK	Sell Limit entry order. ASK >	ASK >	ASK <																																																				
8	BUYSTOP_BY_BID	Buy Stop entry order. BID >	BID <	BID >																																																				
9	SELLSTOP_BY_ASK	Sell Stop entry order. ASK <	ASK >	ASK <																																																				
amount	Mandatory. An amount of the primary currency CCY1.																																																							
price	Optional. If the price is not defined, i.e. price is "0", and side is set to "0" or "1", a market order will be placed. If the price is defined and a side is set from "2" to "9", an entry order will be placed.																																																							
stoploss	Optional. StopLoss level. Value must be defined in pips. Default value is "0", i.e. StopLoss is not set.																																																							
takeprofit	Optional. TakeProfit level. Value must be defined in pips. Default value is "0", i.e. TakeProfit is not set.																																																							

Examples:

Place a market order
"BUY 0.25 MIO EUR/USD @MKT"

```
SubmitOrder ("ps001", "EUR/USD", 0, 0.25, 0, 0, 0);
```

Place a market order with predefined TakeProfit and StopLoss conditions
 "SELL 5 MIO USD/CHF @MKT SL 200 TP 100"

```
SubmitOrder ("ps002", "USD/CHF", 1, 5, 0, 200, 100);
```

Place a BUYLIMIT entry order (ASK <) with a predefined price 1.21
 "BUY 0.27 MIO EUR/USD AT 1.21"

```
SubmitOrder ("or001", "EUR/USD", 2, 0.27, 1.21, 0, 0);
```

Place a SELLSTOP entry order (BID >) with predefined TakeProfit and StopLoss conditions

"SELLSTOP 5 MIO USD/CHF AT 1 SL 200 TP 100"

```
SubmitOrder ("or002", "EUR/USD", 9, 5, 1, 200, 100);
```

3.2 ClosePosition

```
int ClosePosition (string reference);
```

The function is used to close an open position. If the function succeeds, the return value is a number of the closed position. If the function fails, the return value is a [negative number](#).

Parameters	Description
reference	Mandatory. A unique identifier. Maximum 8 characters allowed.

Examples:

Close a position

```
ClosePosition ("ps001");
```

3.3 CloseProfitPosition

```
int CloseProfitPosition (string reference, int profitpips);
```

The function is used to close an open position only in the event that profit for this position is more or equal to defined amount of pips. If the function succeeds, the return value is a number of the closed position. If the function fails, the return value is a [negative number](#).

Parameters	Description
reference	Mandatory. A unique identifier. Maximum 8 characters allowed.
profitpips	Mandatory. An amount of pips.

Examples:

Close a position only in the event that profit for this position is more or equal to 5 pips

```
CloseProfitPosition ("ps001", 5);
```

3.4 CancelOrder

```
int CancelOrder (string reference);
```

The function is used to cancel a submitted entry order. If the function succeeds, the return value is a number of a position in DDS2 Platform that would be opened. If the function fails, the return value is [negative number](#).

Parameters	Description
reference	Mandatory. A unique identifier. Maximum 8 characters allowed.

Examples:

Cancel an order

```
CancelOrder ("or001");
```

3.5 SubmitStop

```
int SubmitStop (string reference, int direction, int pips);
```

The function is used to set a StopLoss or TakeProfit level for an open position or a placed entry order.

Parameters	Description																												
reference	Mandatory. A unique identifier. Maximum 8 characters allowed.																												
direction	Mandatory. <table><tr><th>Direction</th><th>Type</th><th>LONG</th><th>SHORT</th></tr><tr><td>0</td><td>STOPLOSS</td><td>BID <</td><td>ASK ></td></tr><tr><td>1</td><td>TAKEPROFIT</td><td>BID ></td><td>ASK <</td></tr><tr><td>2</td><td>STOPLOSS_BY_ASK</td><td>ASK <</td><td>ASK ></td></tr><tr><td>3</td><td>STOPLOSS_BY_BID</td><td>BID <</td><td>BID ></td></tr><tr><td>4</td><td>TAKEPROFIT_BY_ASK</td><td>ASK ></td><td>ASK <</td></tr><tr><td>5</td><td>TAKEPROFIT_BY_BID</td><td>BID ></td><td>BID <</td></tr></table> <p>If direction is set to “0” or “1”, StopLoss (SL) and TakeProfit (TP) will be set automatically according to a side of order or position.</p>	Direction	Type	LONG	SHORT	0	STOPLOSS	BID <	ASK >	1	TAKEPROFIT	BID >	ASK <	2	STOPLOSS_BY_ASK	ASK <	ASK >	3	STOPLOSS_BY_BID	BID <	BID >	4	TAKEPROFIT_BY_ASK	ASK >	ASK <	5	TAKEPROFIT_BY_BID	BID >	BID <
Direction	Type	LONG	SHORT																										
0	STOPLOSS	BID <	ASK >																										
1	TAKEPROFIT	BID >	ASK <																										
2	STOPLOSS_BY_ASK	ASK <	ASK >																										
3	STOPLOSS_BY_BID	BID <	BID >																										
4	TAKEPROFIT_BY_ASK	ASK >	ASK <																										
5	TAKEPROFIT_BY_BID	BID >	BID <																										
pips	Mandatory. In case of “0” for direction “0” or “1”, the function will remove a condition.																												

Examples:

Define a new TakeProfit value for an open position

```
SubmitStop ("ps002", 1, 100);
```

Delete TakeProfit for an open position

```
SubmitStop ("ps002", 1, 0);
```

Define a new StopLoss by BID value for an entry order

```
SubmitStop ("or002", 3, 100);
```

Delete TakeProfit for an entry order

```
SubmitStop ("or002", 5, 0);
```

3.6 PlaceOffer

```
int PlaceOffer (string reference, string instruments, int side, double amount, double price, int duration);
```

The function is used to place an entry order into the "Market depth". The function returns a number of the opened position in DDS2 Platform or a number of a position in DDS2 Platform that would be opened or [error code](#) in case of failure.

Parameters	Description
------------	-------------

reference	Mandatory. A unique identifier. Maximum 8 characters allowed.		
instruments	Mandatory. Instruments for trading. CCY1/CCY2.		
side	Mandatory. Type of operation.		
	Side	Type	Description
	0	BUY	Place Bid entry order.
	1	SELL	Place Offer entry order.
amount	Mandatory. An amount of the primary currency CCY1.		
price	Mandatory. Placing order with price worse then market price level causes immediate execution at market price. Orders placing into the "Market depth", also effects on your available margin, depending on your order size and leverage.		
duration	Mandatory. Execution time of order. Value must be defined in minutes. If the duration is "0", GTC (Good Till Cancel) order will be placed. In others case Good For (valid for the specified period) order will be opened.		

4. Error codes

Error code	Description
-1	Receive error.
-2	Connection error. DDS2 client is not launched.
-3	Type error.
-4	Reference parameter exceeded 8 characters.
-10	Type of order unknown.
-11	Timeout. Order cannot be processed more than 30 sec.
-12	DDS2 exception.
-13	ID not found. Provided label is not found.
-14	ID not unique. Provided ID already registered.
-15	The calculated price for StopLoss or TakeProfit condition is a negative value.
-16	Incorrect CCY1 amount.
-17	No liquidity at this price.
-18	Duration is a negative value.
-99	Unknown error.

Appendix A: MQL4

1. Include file for MetaTrader (MetaTrader 4_HOME\experts\include\DDS2Agent.mqh)

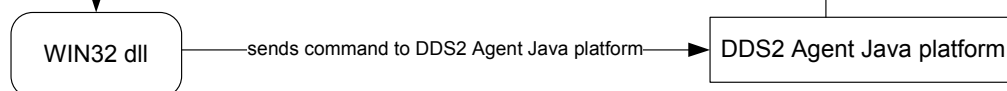
```
#import "DDS2Agent.dll"
int SubmitOrder (string reference, string instruments, int side,
double amount, double price, int stoploss, int takeprofit);
int ClosePosition (string reference);
int CloseProfitPosition (string reference, int profitpips);
int CancelOrder (string reference);int SubmitStop (string
reference, int direction, int pips);
int SubmitStop (string reference, int direction, int pips);
int PlaceOffer (string reference, string instruments, int side,
double amount, double price, int duration);
```

2. Example of DLL calling from MetaTrade 4 (MQL4)

```
int ddsOrderOpen(string reference, string instruments, int side, double amount, double price, int
stoploss, int takeprofit){
    int ticket_dds = SubmitOrder(reference, instruments, side, amount, price, stoploss, takeprofit);
    if(ticket_dds < 1){
        Print("Error: position was not opened. Code (\"+ticket_dds+\")");
    } else{
        Print("Order ID = ",ticket_dds);
        double price = Ask;
        if(command==OP_SELL){
            price = Bid;
        }
        int ticket_ts = OrderSend(instrument,side,amount,price,3,0,0,"expert comment",255,0,Green);
        if(ticket_ts < 1){
            int error = GetLastError();
            Print("Error = ",ErrorDescription(error));
            return;
        }
    }
    return(ticket_dds);
}
```

sends command to external dll

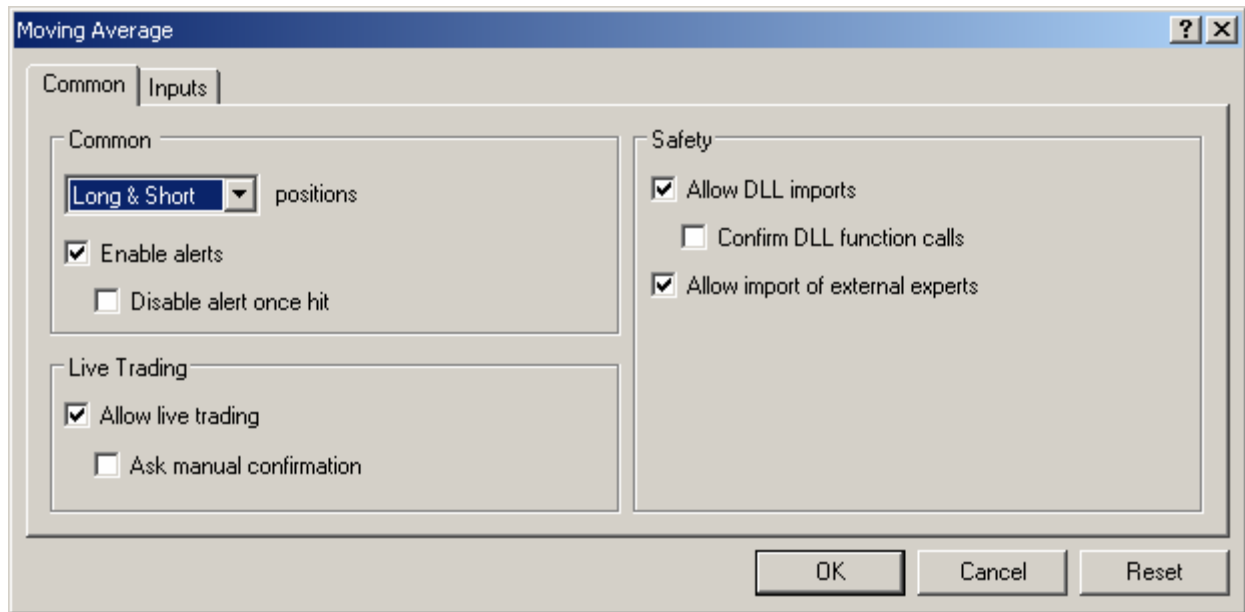
returns code to 3rd Strategy platform



3. Configuration of MetaTrader 4

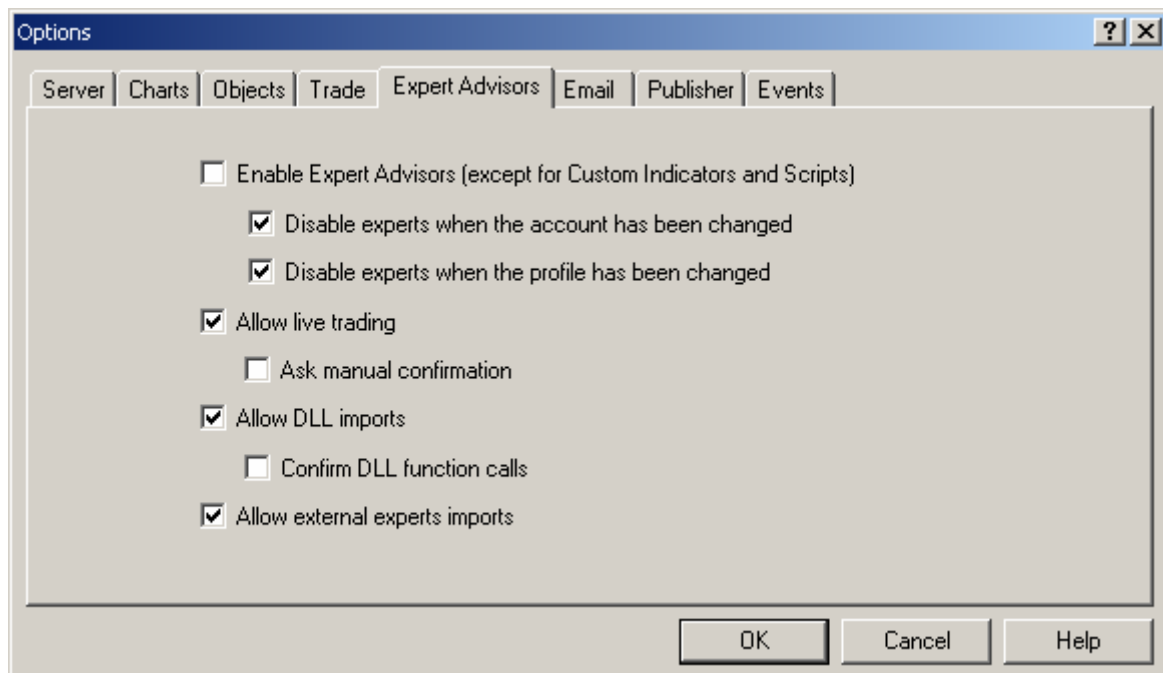
Scripts

Check Tools \ Options (or just press Ctrl+O) “Allow live trading” and “Allow DLL imports” checkboxes.



Expert Advisors

Attach to a chart Expert Advisors strategy or click twice on the selected strategy. In dialog window, check “Allow live trading” and “Allow DLL imports” checkboxes.



Appendix B: EasyLanguage

```
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll",
int, "SubmitOrder", lpstr, lpstr, int, double, double, int, int;
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll", int, "
ClosePosition", lpstr;
```

```
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll", int, "  
CloseProfitPosition", lpstr, int;  
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll", int, "  
CancelOrder", lpstr;  
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll", int, "  
SubmitStop", lpstr, int, int;  
DefinedDLLFunc: "C:\WINDOWS\system32\DDS2Agent.dll", int, "  
PlaceOffer", lpstr, lpstr, int, double, double, int;
```