**How to install**

1. Download installation file from mtapi4.net

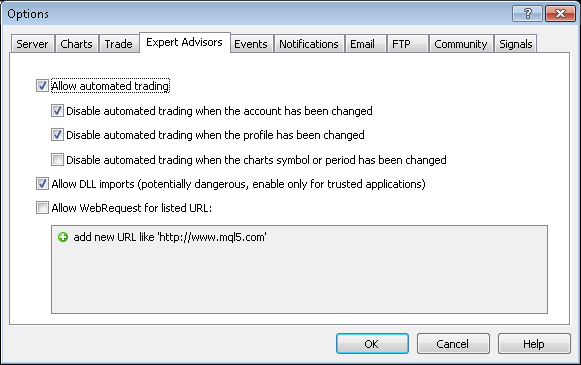
MtApi5 installation file (32 bit): MtApi5Setup\_x86\_1.0.5.msi

MtApi5 installation file (64 bit): MtApi5Setup\_x64\_1.0.5.msi

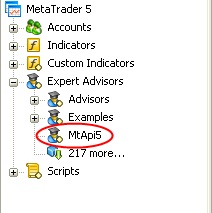
1. Install MtApi5 to program's folder (by default "C:\Program Files").
2. Install MtApi5 expert into the terminal:

Click on MtApi5.ex5 from "Start -> MtApi5 -> Expert". Expert will be added to the terminal automatically.

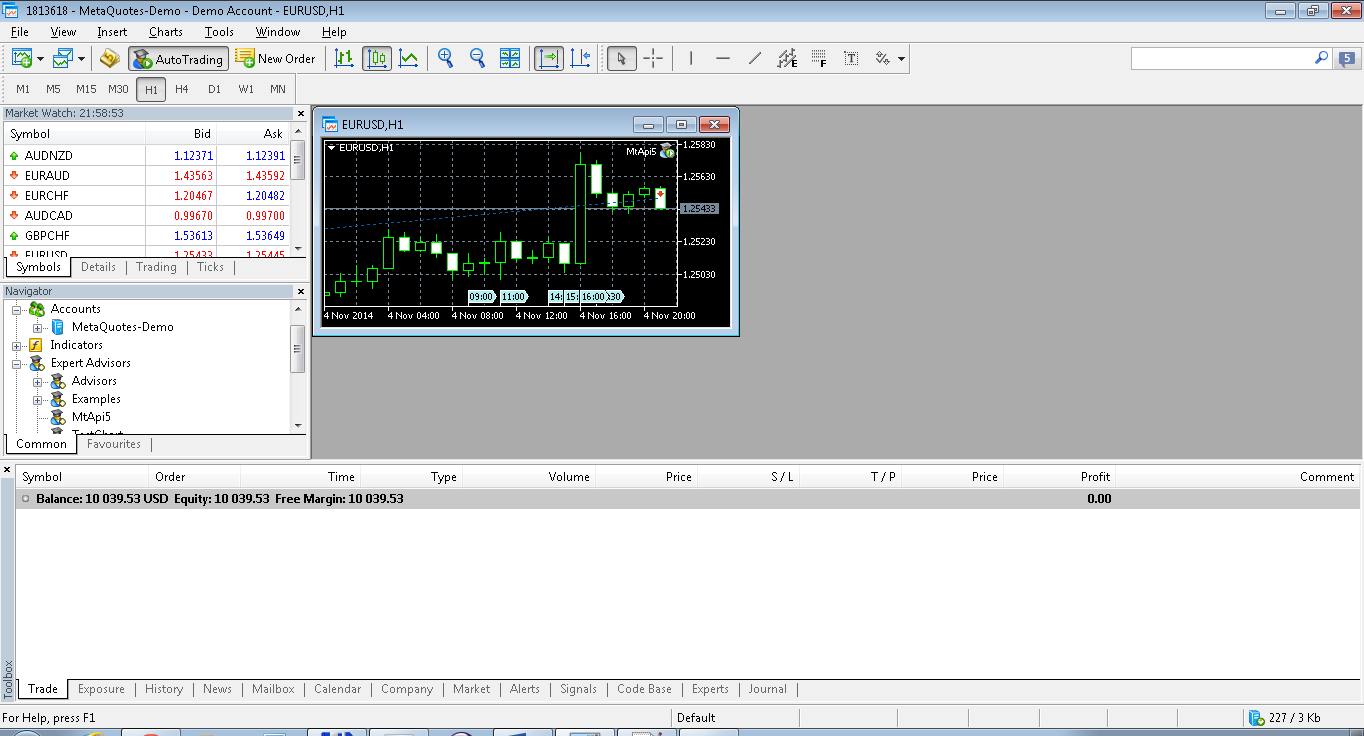
1. Setup expert's options:



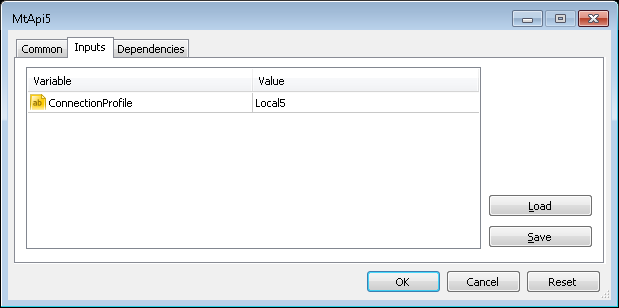
1. Attach MtApi5 Expert Advisor to Chart.



MtApi5 Expert will be shown in right top corner of chart window:



1. Set parameters of MtApi Expert:



Variable ConnectionProfile defines a name of connection profile.

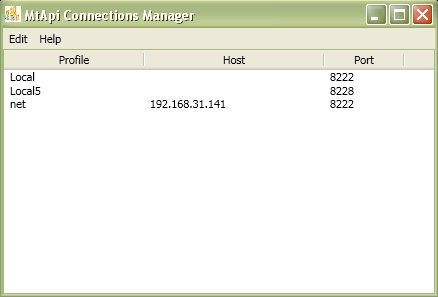
It has default values: Local5.

Profile "Local5" has port 8228 and can be used only for local connection (terminal and client on one machine).

To create different profiles use application **ConnectionsManager**:

Start🡪MtApi5🡪 ConnectionsManager.

User can create several different connections to one terminal.



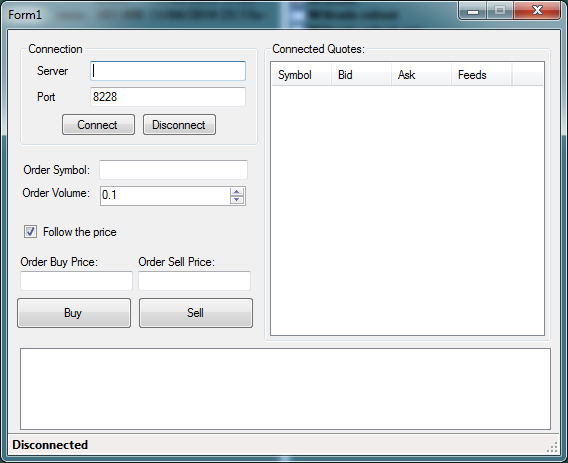
All profiles are saved in registry

"HKEY\_CURRENT\_USER\Software\MtApi\ConnectionProfiles".

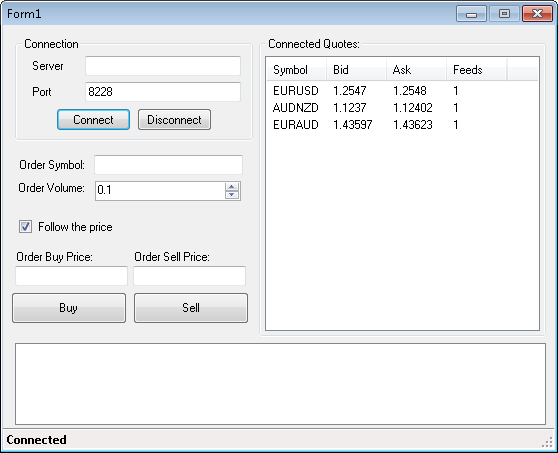
**How to use binary files of test application**

1. Run binary file from folder

\Mt5trade\Mt5trade\bin\Release\Mt5trade.exe

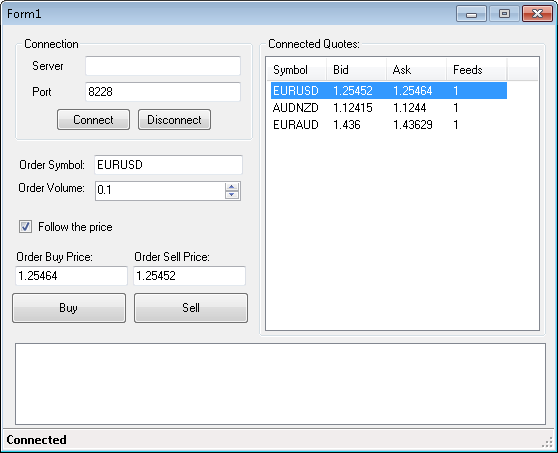


1. Click on button “Connect”. Application will connect to terminal.

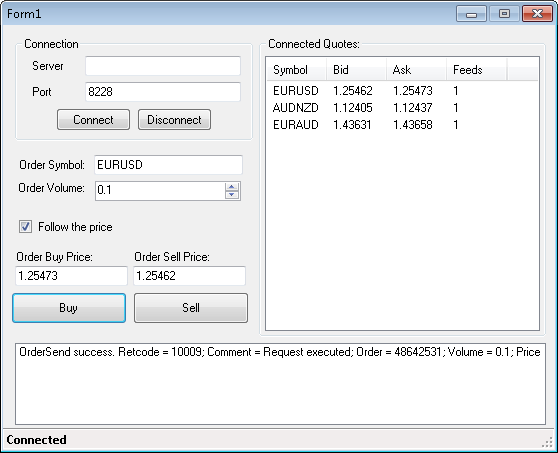


1. Enter Order Symbol, Order Volume and Order Price.

Or select quote from Connected Quotes list to fill order fields automatically.

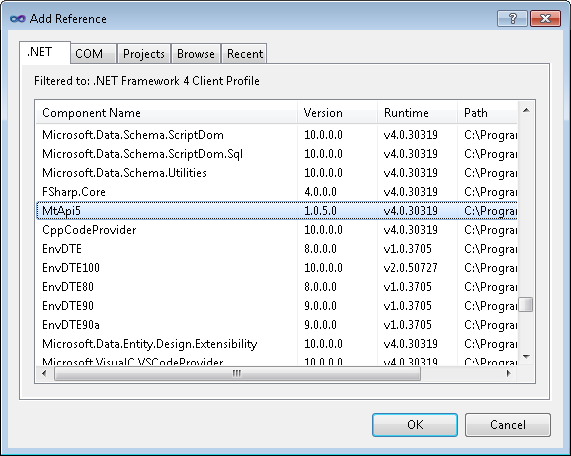


1. Press button Buy or Sell to send order.



**How to use Visual Studio project**

1. Open project Mt5trade in Visual Studio 2010 (Mt5trade.sln).
2. Add reference to MtApi5 library into project Mt5Trade



1. Build project and run test application

**Examples of source code:**

1. Create object of API

private MtApi5Client apiClient = new MtApi5Client();

1. Subscribe to Api events

apiClient.QuoteAdded += apiClient\_QuoteAdded;

apiClient.QuoteRemoved += apiClient\_QuoteRemoved;

apiClient.QuoteUpdated += apiClient\_QuoteUpdated;

apiClient.ConnectionStateChanged += apiClient\_ConnectionStateChanged;

1. Send order

var request = new MqlTradeRequest { Action = ENUM\_TRADE\_REQUEST\_ACTIONS.TRADE\_ACTION\_DEAL

, Symbol = symbol

, Type = orderType

, Price = price

, Volume = volume

, Comment = "Test Trade Request"

};

MqlTradeResult result;

bool retVal = apiClient.OrderSend(request, out result);