



# Integrating SQL Server 2008 with Metatrader 5



**Kanhar Munshi**

4 Mar 2010 CPOL

Invoking an External ADO.NET DLL to connect and execute queries on a SQL Server database from within the Meta Trader Platform

[Download source - 108.64 KB](#)

## Introduction

There is only one peer reviewed article concerning SQL integration with Metatrader by [Yuriv Zaytsev](#) written almost 8 years back and which was written originally in Russian and automatically translated to English using a translator. The example is very hard to execute, and requires creating a specific database, with specific login credentials and roles without which a successful result cannot be achieved. The article itself can be found [here](#). [Zaytsev](#) used the following:

- Microsoft SQL SERVER 2000 Developer - BASE
- VISUAL C++ 6.0 SP5 - to create DLL "YZMSSQLExpertSample.dll"
- MDAC 7

Furthermore his example is specific to a database, which you have to create, and connection strings that you have to hardcode. His article was also written when Meta trader 5 was not around, and I had considerable compatibility problems with upgrading from MQL4 to MQL5 using his code.

The article's code samples do not easily upgrade to Visual Studio .NET versions 2003 or 2008, so this is an attempt to remedy all of the above providing a more seamless implementation background.

The DLL has also been modified to accept a connection string as an argument thereby removing any need to constantly rebuild the DLL for connection string changes, and can also accept an entire SQL statement as an input using its **ExecuteScalar** method.

## Background

This article is very relevant to people attempting to integrate the newer Metatrader 5 terminal with SQL Server 2008 to receive and store in a database, commodities, and currencies real time streaming data, to build associating trading systems.

## Using the Code

The code has three parts:

**Part 1:** DLL Generation using the *DLL\_Creator.zip* VS 2008 DLL Project. (No changes are necessary here). However you must ensure that your *MSADO15.dll* is in the location "C:\Program Files\Common Files\System\ADO\msado15.dll". If it's not, you could either manually place it there or rebuild the DLL project with the correct path.

**Part 2:** Importing the generated DLL into the *Libraries* folder. On my computer, the location for this is `C:\Users\Administrator\AppData\Roaming\MetaQuotes\Terminal\5D2A9C702A29311ED87B6AD8A346121B\MQL5\Libraries\`.

But to be sure, on MT5 you can just click on File->Open Data Folder and then Click on Libraries and place the DLL file inside.

**Note:** Make sure you check the "Allow DLL Imports" Box as shown below:



**Part 3:** Create a new MQL5 script with only the following code that references the DLL.

C++

```
#import "ExpertSample.dll"
    string ExecuteScalar(string strSQL, string strCNN);
#import

int OnStart()
{
    string temp = SymbolInfoDouble("EURUSD",SYMBOL_BID);
    ExecuteScalar("INSERT INTO [indicator_data]([indicator_id] ,[value])
    VALUES (2, "+temp +");", "Provider=SQLOLEDB.1;User ID=sa;
    Password=*****;Persist Security Info=True;
    Initial Catalog=Finance;Data Source=KMS\\SQL2008" );
    return(0);
}
```

## Points of Interest

Well I spent a good couple of days trying to understand why passing strings from MQL5 to C++ only passed the first character. I published this as a bug in [mql5's ability to pass variables via dll](#). I was told that MQL5 handles strings as Wide Characters or `w_char`. I was receiving them as `char*` instead of `w_char *` and that made all the difference. The code is reproduced below:

C++

```
void p(char *s)
{
    MessageBox(NULL, s, (LPCSTR)L"no error",MB_ICONWARNING |
    MB_CANCELTRYCONTINUE | MB_DEFBUTTON2 );
}
```

```

MT4_EXPFUNC char * __stdcall ExecuteScalar(wchar_t *wstr, wchar_t *wcnn)
{
    char* sql = new char[wcslen(wstr) + 1];
    wcstombs( sql, wstr, wcslen(wstr) );
    sql[wcslen(wstr)] = '\\0';
    //p(sql);
    p(sql);
    char *cnn = new char[wcslen(wcnn) + 1];
    wcstombs(cnn, wcnn, wcslen(wcnn) );
    cnn[wcslen(wcnn)] = '\\0';
    //p(cnn);

    HRESULT hr = S_OK;
    char tmpChar[255];
    try {
        // Define string variables.

        //_bstr_t strCnn("Provider=SQLOLEDB.1;User ID=sa;
        //Password=Admin1234;Persist Security Info=True;
        //Initial Catalog=Finance;Data Source=KMS\\SQL2008");
        _bstr_t strCnn(cnn);

        ::CoInitialize(NULL);
        _RecordsetPtr pRstAuthors = NULL;
        // Call Create instance to instantiate the Record set
        hr = pRstAuthors.CreateInstance(__uuidof(Recordset));
        if(FAILED(hr))
        {
            ::CoUninitialize();
            p("ERROR: Failed creating record set instance");
            return "ERROR: Failed creating record set instance";
        }

        //Open the Record set for getting records from Author table
        try {
            pRstAuthors->Open(sql,strCnn, adOpenStatic,      adLockReadOnly,adCmdText);
            //int msgboxID2 = MessageBox(NULL,(LPCSTR)L"no errors",
            //((LPCSTR)L"no error",MB_ICONWARNING |
            //MB_CANCELTRYCONTINUE | MB_DEFBUTTON2 );
        } catch (_com_error & ce1) {

            ::CoUninitialize();
            p("Unable to Open SQL Server");
            //int msgboxID = MessageBox(NULL,(LPCSTR)
            //"Resource not available\\nDo you want to try again?",
            //((LPCSTR)L"SQL Error",MB_ICONWARNING | MB_CANCELTRYCONTINUE |
            //MB_DEFBUTTON2 );
            return "ERROR: Unable to open SQL Server";
        }
        p("SQL Opened");

        try {
            pRstAuthors->MoveFirst();
        } catch(_com_error) {
            ::CoUninitialize();
            return ""; //empty data
        }

        //Loop through the Record set
        if (!pRstAuthors->EndOfFile)
        {
            _variant_t tmpvariant;
            //Get the first column value
            tmpvariant = pRstAuthors->GetFields()->GetItem((long)0)->GetValue();

```

```

        strcpy(tmpChar, (_bstr_t)tmpvariant);
    }

    if (pRstAuthors->State == adStateOpen)
        pRstAuthors->Close();

    pRstAuthors = NULL;
    ::CoUninitialize();
}
catch(_com_error & ce)
{
    //_bstr_t strError = ce.ErrorMessage;
    ::CoUninitialize();
    p("ERROR: Failed to get data.");
    return "ERROR: Failed to get data.";
}

return tmpChar;
}

```

## Conclusion

To get this guy running, all you need is a database to test on and a MQL 5 platform available at [MetaTrader5](#).

You will also need [VS2008](#) the express versions of which are free, in case you want to change the DLL file.

This is based on the smash and grab concept, and implementing this should be easy. Good luck and feel free to post comments and I will do my best to reply.

## History

- 4<sup>th</sup> March, 2010: Initial post

## License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

## About the Author



### Kanhar Munshi



Software Developer (Senior) Microsoft  
United States 🇺🇸

**\*\*Update\*\*:** I now work at Microsoft. See more here: <https://kanhar.github.io/>

\*Update\*: I now work as a software developer at Investment Bank, working on mostly proprietary stuff.

Kanhar Munshi currently works as a consultant for the New York City Department of Health and Environmental Surveillance building, deploying and maintaining ASP.NET web applications, while also serving in the role of an assistant database administrator.

He is currently involved in the building of a Large Scale implementation of a Normalized Data Store, Data Warehouse for the Department of Health.

His interests, include C#, running, table tennis, triathlons and going on long day hikes.

## Comments and Discussions

 **12 messages** have been posted for this article Visit <https://www.codeproject.com/Articles/60371/Integrating-SQL-Server-2008-with-Metatrader-5> to post and view comments on this article, or click [here](#) to get a print view with messages.

[Permalink](#)  
[Advertise](#)  
[Privacy](#)  
[Cookies](#)  
[Terms of Use](#)

Article Copyright 2010 by Kanhar Munshi  
Everything else Copyright © [CodeProject](#), 1999-2021

Web01 2.8.20210820.1