
Subject: Re: MS SQL Server connection possible?
Posted by [DrGary](#) on Thu, 08 Mar 2007 22:25:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Success! Here's how I connected to a MS SQL Server from Windows. Please comment/advise. Is there a better/simpler way to code SQL to SQL Server?

The code is copied from Example 1 of:
<http://cs.ua.edu/components/integrated/handouts/08-odbc-example.pdf>
and compiled with MINGW.

What I did:

- 1) Find "Administrative Tools | Data Sources (ODBC)". It may be in Control Panels, but can also be on your Start menu depending on how you set up Windows. "Add..." an "SQL Server" driver and fill out the connection details. The "Name:" you give to the datasource is the one you'll use in the code. I selected "With SQL Server authentication using a login ID and password"; putting in a login and password allowed the "Test Data Source..." button on the last screen to test the connection. Everything else, I left at the default settings.
- 2) In TheIDE, File | "Set Main Package..", click "New package". I created a "Console application (no U++)".
- 3) Paste the code below as the main cpp file. Change the SQLConnect statement at line 72 to use the name of the data source connection you created in step 1. Change the name and password in line 81.
- 4) In the new package, select Project | "Package organizer..."; right click in the big window, select "New Link options.."; leave the "When" field blank, add "-lodbc32" without quotes to the field to the right of the ":".
- 5) Control-F5 to run. It continuously hits the database, so hit Control-c to stop.

```
//http://cs.ua.edu/components/integrated/handouts/08-odbc-example.pdf
// Include The Appropriate Header Files
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <iostream.h>
#include <stdio.h>

// Define The ODBC_Class Class
class ODBC_Class {
    // Attributes
```

```

public:
    SQLHANDLE EnvHandle;
    SQLHANDLE ConHandle;
    SQLHANDLE StmtHandle;
    SQLRETURN rc;
    // Operations
public:
    ODBC_Class(); // Constructor
    ~ODBC_Class(); // Destructor
    SQLRETURN ShowResults();
};

// Define The Class Constructor
ODBC_Class::ODBC_Class() {
    // Initialize The Return Code Variable
    rc = SQL_SUCCESS;
    // Allocate An Environment Handle
    rc = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &EnvHandle);
    // Set The ODBC Application Version To 3.x
    if (rc == SQL_SUCCESS)
        rc = SQLSetEnvAttr(EnvHandle, SQL_ATTR_ODBC_VERSION,
                           (SQLPOINTER) SQL_OV_ODBC3, SQL_IS_INTEGER);
    // Allocate A Connection Handle
    if (rc == SQL_SUCCESS)
        rc = SQLAllocHandle(SQL_HANDLE_DBC, EnvHandle, &ConHandle);
}

// Define The Class Destructor
ODBC_Class::~~ODBC_Class() {
    // Free The Connection Handle
    if (ConHandle != NULL)
        SQLFreeHandle(SQL_HANDLE_DBC, ConHandle);
    // Free The Environment Handle
    if (EnvHandle != NULL)
        SQLFreeHandle(SQL_HANDLE_ENV, EnvHandle);
}

// Define The ShowResults() Member Function
SQLRETURN ODBC_Class::ShowResults() {
    // Declare The Local Memory Variables
    SQLINTEGER val;
    // Bind The Columns In The Result Data Set Returned To
    // Application Variables
    rc = SQLBindCol(StmtHandle, 1, SQL_C_SLONG, (SQLPOINTER) &val,
                   sizeof(val), NULL);
    rc = SQLFetch(StmtHandle);
    if (rc != SQL_NO_DATA){
        cout << "Returned value " << val << endl;
    }
}

```

```

    } else
        cout << "No data returned." << endl;
    // Return The ODBC API Return Code To The Calling Function
    return(rc);
}

/*-----*/
/* The Main Function */
/*-----*/
int main()
{
    // Declare The Local Memory Variables
    SQLRETURN rc = SQL_SUCCESS;
    SQLCHAR DBName[14] = "obdcsourcename";
    SQLCHAR SQLStmt[255];
    char * title;
    //title = new char[255];
    // Create An Instance Of The ODBC_Class Class
    ODBC_Class Example;
    // Connect To The Northwind Sample Database
    if (Example.ConHandle != NULL) {
        rc = SQLConnect(Example.ConHandle, DBName, SQL_NTS,
            (SQLCHAR *) "user", SQL_NTS, (SQLCHAR *) "password", SQL_NTS);
        while (1){
            // Allocate An SQL Statement Handle
            rc = SQLAllocHandle(SQL_HANDLE_STMT, Example.ConHandle,
                &Example.StmtHandle);
            if (rc == SQL_SUCCESS) {
                strcpy((char *) SQLStmt, "SELECT 1");
                // Prepare And Execute The SQL Statement
                rc = SQLExecDirect(Example.StmtHandle, SQLStmt, SQL_NTS);
                // Display The Results Of The SQL Query
                if (rc == SQL_SUCCESS) {
                    cout << "Query execution success." << endl;
                    Example.ShowResults();
                } else {
                    cout << "Query execution failure." << endl;
                }
            }
            // Free The SQL Statement Handle
            if (Example.StmtHandle != NULL)
                SQLFreeHandle(SQL_HANDLE_STMT, Example.StmtHandle);
        }
        // Disconnect From The Northwind Sample Database
        rc = SQLDisconnect(Example.ConHandle);
    }
    // Return To The Operating System
    return(rc);
}

```

}