

How to convert XLS to SQL server in C# and ByteScout Spreadsheet SDK

How to code in C# to convert XLS to SQL server with this step-by-step tutorial

This sample source code below will demonstrate you how to convert XLS to SQL server in C#. Want to convert XLS to SQL server in your C# app? ByteScout Spreadsheet SDK is designed for it. ByteScout Spreadsheet SDK is the SDK that can write and read, modify and calculate Excel and CSV spreadsheets. Most popular formulas are supported. You may import or export data to and from CSV, XML, JSON as well as to and from databases, arrays.

This rich sample source code in C# for ByteScout Spreadsheet SDK includes the number of functions and options you should do calling the API to convert XLS to SQL server. In your C# project or application you may simply copy & paste the code and then run your app! Detailed tutorials and documentation are available along with installed ByteScout Spreadsheet SDK if you'd like to dive deeper into the topic and the details of the API.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Spreadsheet SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Spreadsheet SDK](#)

[Get Free API key for Web API](#)

[visit www.Bytescout.com](http://www.Bytescout.com)

Source Code Files:

```

using System;
using System.IO;
using Bytescout.Spreadsheet;
using System.Data.SqlClient;

namespace ExportToSQLServer
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // MODIFY THE CONNECTION STRING WITH YOUR CREDENTIALS!
                string connectionString = "Data Source=localhost;Initial Catalog=TestDB;User ID=sa;Password=1qaz!@WSX";

                using (SqlConnection connection = new SqlConnection(connectionString))
                {
                    connection.Open();

                    // Drop test database if exists
                    ExecuteQueryWithoutResult(connection, "DROP DATABASE TestDB");

                    // Create empty database
                    ExecuteQueryWithoutResult(connection, "CREATE DATABASE TestDB");
                    // Switch to created database
                    ExecuteQueryWithoutResult(connection, "USE TestDB");
                    // Create a table for XLS data
                    ExecuteQueryWithoutResult(connection, "CREATE TABLE TestTable (ID INT, Name VARCHAR(50))");

                    // Load XLS document
                    using (Spreadsheet document = new Spreadsheet("SimpleReport.xls"))
                    {
                        document.LoadFromFile("SimpleReport.xls");
                        Worksheet worksheet = document.Workbook.Worksheets[0];

                        for (int row = 0; row <= worksheet.UsedRange.Rows - 1; row++)
                        {
                            String insertCommand = string.Format("INSERT INTO TestTable (ID, Name) VALUES ({0}, '{1}');",
                                worksheet.Cell(row, 0).Text,
                                worksheet.Cell(row, 1).Text);
                            ExecuteQueryWithoutResult(connection, insertCommand);
                        }
                    }

                    // Check the data successfully exported
                    using (SqlCommand command = new SqlCommand("SELECT * FROM TestTable", connection))
                    {
                        SqlDataReader reader = command.ExecuteReader();

                        if (reader != null)
                        {
                            Console.WriteLine();
                            Console.WriteLine("Exported XLS data to SQL Server successfully.");
                            Console.WriteLine();
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }
    }
}

```

```

        while (reader.Read())
        {
            Console.WriteLine(reader.GetString(0));
        }
    }

    Console.WriteLine();
    Console.WriteLine("Press any key.");
    Console.ReadKey();
}

}

catch (Exception ex)
{
    Console.WriteLine("Error: " + ex.Message);
    Console.ReadKey();
}

}

static void ExecuteQueryWithoutResult(SqlConnection connection, string query)
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.ExecuteNonQuery();
    }
}

}

}
```

VIDEO

https://www.youtube.com/watch?v=nm_7I0PN1TY

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Spreadsheet SDK Home Page](#)
[Explore ByteScout Spreadsheet SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Spreadsheet SDK Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)

[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

[visit \[www.PDF.co\]\(http://www.PDF.co\)](#)

[\[www.bytescout.com\]\(http://www.bytescout.com\)](#)