

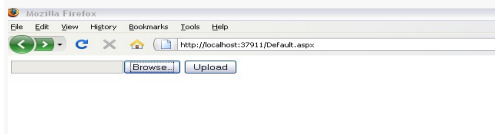
Import data from EXCEL to SQL in ASP.NET

Published 6/13/2013 by [Raghav Khunger](#)

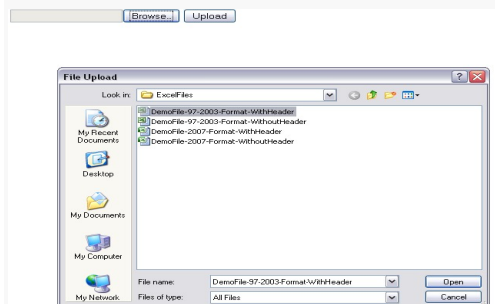
In this article I will explain how to import data from EXCEL to SQL in ASP.NET . In many situations we have data in the form of excel sheet but we have the requirement to have that data in SQL SERVER DB. I have explained importing data both from **Excel 97-2003 as well as Excel 2007 format**.

Straight way cutting the crap I am coming to the practical. Here are the steps that you need to do in order to test the code given below in this article. Run the project HowToImportExcelData which is present in the rar file attached with this article below.

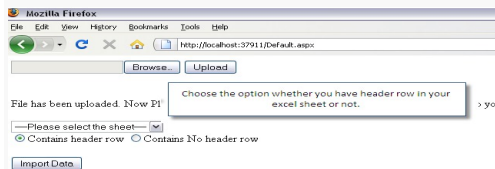
- On running the default page you will see the below page.



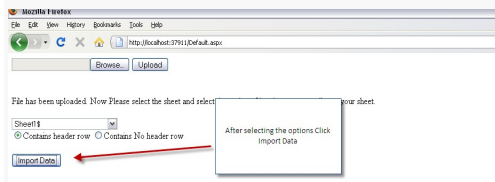
- Click the Upload button.



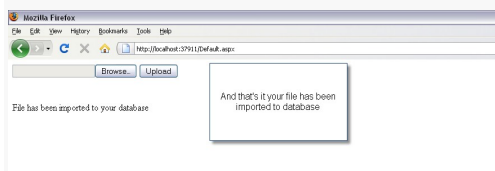
- Select the sheet which you want to import. Also select the option whether you have header row in your excel sheet or not.



- Click the *ImportData* button.



- And that's it your data has imported to database.



Now come to the code which actually makes these things happen.

In SQL

```
CREATE TABLE [TestImportFromExcel]
(
    [ID] INT,
    [FirstName] NVARCHAR(100),
    [LastName] NVARCHAR(100),
    [Age] INT,
    [Sex] NVARCHAR(10)
)

GO

CREATE PROCEDURE [USP_ImportFromExcel1972003Format]
(
    @DATASOURCE NVARCHAR(200),
    @HASHEADER NVARCHAR(10),
    @SHEET NVARCHAR(10)
)
AS
BEGIN

    DECLARE @SQL NVARCHAR(MAX)

    SET @SQL='INSERT INTO [TestImportFromExcel]
SELECT * FROM OPENDATASOURCE
(''Microsoft.Jet.OLEDB.4.0'', ''Data Source=' + @DATASOURCE+';
Extended Properties='''Excel 8.0;HDR='+ @HASHEADER+''''''')...'+@SHEET

EXEC SP_EXECUTESQL @SQL

END

GO

ALTER PROCEDURE [USP_ImportFromExcel12007Format]
(
    @DATASOURCE NVARCHAR(200),
    @HASHEADER NVARCHAR(10),
    @SHEET NVARCHAR(10)
)
AS
BEGIN

    DECLARE @SQL NVARCHAR(MAX)

    SET @SQL='INSERT INTO [TestImportFromExcel]
SELECT [ID],[FirstName], [LastName], [Age], [Sex] FROM OPENDATASOURCE
(''Microsoft.Ace.OLEDB.12.0'', ''Data Source=' + @DATASOURCE+';
Extended Properties='''Excel 12.0;HDR='+ @HASHEADER+''''''')...'+@SHEET

EXEC SP_EXECUTESQL @SQL

END

--WE NEED TO EXECUTE THE BELOW TWO BATCH FOR FIRST TIME
--IF WE WANT TO USE OPENDATASOURCE OR OPENROWSET QUERY THAN WE MUST HAVE TO ENABLE
--"AD HOC DISTRIBUTED QUERIES".
--IT IS DISABLED BY DEFAULT AS A PART OF SECURITY CONFIGURATION

GO
SP_CONFIGURE 'show advanced options', 1
RECONFIGURE
GO

SP_CONFIGURE 'Ad Hoc Distributed Queries', 1
RECONFIGURE
```

In Aspx

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
```

```

<body>
<form id="form1" runat="server">
<div id="divBeforeFileUpload" runat="server">
<asp:FileUpload runat="server" ID="myFileUpload" />
<asp:Button runat="server" Text="Upload" ID="btnUpload" OnClick="btnUpload_Click" /><br />
<br />
</div>
<div id="divAfterFileUpload" runat="server" visible="false" style="float: left;">
<asp:Label runat="server" Text="" ID="lblFileUploadedStatus"></asp:Label><br /><br />
<asp:DropDownList runat="server" ID="ddlNoOfSheets">
</asp:DropDownList>
<br />

<asp:RadioButton runat="server" ID="optHasHeader" Text="Contains header row" GroupName="SheetHeader"
Checked="true" />
<asp:RadioButton runat="server" ID="optNoHeader" Text="Contains No header row " GroupName="SheetHeader" />
<br />
<br />
<asp:Button runat="server" Text="Import Data" ID="btnImportData" OnClick="btnImportData_Click" />
</div>
<div id="divAfterFileImport" runat="server" visible="false" style="float: left;">
<asp:Label runat="server" Text="" ID="lblImportData"></asp:Label><br />
</div>
</form>
</body>
</html>

```

In CodeBehind

```

using System;
using System.Data;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.IO;
using System.Web.UI.WebControls;

namespace HowToImportExcelData
{
    /// <summary>
    ///
    /// </summary>
    public partial class _Default : System.Web.UI.Page
    {
        private string MyExcel972003FormatCs =
            "Provider=Microsoft.Jet.OLEDB.4.0;Extended Properties='Excel 8.0;HDR={0}'; Data Source={1}";
        private string MyExcel2007FormatCs =
            "Provider=Microsoft.ACE.OLEDB.12.0;Extended Properties='Excel 8.0;HDR={0}'; Data Source={1}";
        private string FileUploadPath = "ExcelFiles";

        /// <summary>
        /// Handles the Click event of the btnUpload control.
        /// </summary>
        /// <param name="sender">The source of the event.</param>
        /// <param name="e">The <see cref="System.EventArgs"/> instance containing the event data.</param>
        protected void btnUpload_Click(object sender, EventArgs e)
        {
            if (myFileUpload.HasFile)
            {
                string filePath, ext;
                UploadFile(out filePath, out ext);
                BindSheetNo(filePath, ext, "Yes");
                divAfterFileUpload.Visible = true;
                lblFileUploadedStatus.Text = "File has been uploaded. Now Please select the sheet and select the option of header row according to your sheet.";
            }
        }

        /// <summary>
        /// Uploads the file.
        /// </summary>
        /// <param name="filePath">The file path.</param>
        /// <param name="ext">The ext.</param>
        protected void UploadFile(out string filePath, out string ext)
        {
            string fileOnClient = myFileUpload.PostedFile.FileName;
            ext = Path.GetExtension(fileOnClient);
            filePath = Server.MapPath(string.Format("{0}/{1}{2}", FileUploadPath, Guid.NewGuid(), ext));
            ViewState["filePath"] = filePath;
            myFileUpload.SaveAs(filePath);
        }

        /// <summary>
        /// Gets the file connection string.
        /// </summary>
        /// <param name="filePath">The file path.</param>
        /// <param name="ext">The ext.</param>
        /// <param name="hasHeaderRow">The has header row.</param>
        /// <returns></returns>
        protected string GetFileConnectionString(string filePath, string ext, string hasHeaderRow)

```

```

{
    string myCs = string.Empty;
    switch (ext)
    {
        case ".xlsx":
            myCs = string.Format(MyExcel2007FormatCs, hasHeaderRow, filePath);
            break;

        case ".xls":
            myCs = string.Format(MyExcel1972003FormatCs, hasHeaderRow, filePath);
            break;
    }
    return myCs;
}

/// <summary>
/// Binds the sheet no.
/// </summary>
/// <param name="filePath">The file path.</param>
/// <param name="ext">The ext.</param>
/// <param name="hasHeaderRow">The has header row.</param>
protected void BindSheetNo(string filePath, string ext, string hasHeaderRow)
{
    string myCs = GetFileConnectionString(filePath, ext, hasHeaderRow);
    if (!string.IsNullOrEmpty(myCs))
    {
        using (OleDbConnection connection = new OleDbConnection(myCs))
        {
            connection.Open(); //Open the connection

            //Bind ddlNoOfSheets with sheet numbers
            ddlNoOfSheets.DataSource = connection.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, null);
            ddlNoOfSheets.DataTextField = "Table_Name";
            ddlNoOfSheets.DataValueField = "Table_Name";
            ddlNoOfSheets.DataBind();

            //Insert item at first position
            ddlNoOfSheets.Items.Insert(0, new ListItem("----Please select the sheet----", "0"));
        }
    }
}

/// <summary>
/// Gets the SQL command.
/// </summary>
/// <returns></returns>
protected string GetSqlCommand()
{
    string myCommand = string.Empty;
    if (ViewState != null)
        switch (Path.GetExtension(ViewState["filePath"].ToString()))
        {
            case ".xlsx":
                myCommand = "USP_ImportFromExcel2007Format";
                break;

            case ".xls":
                myCommand = "USP_ImportFromExcel1972003Format";
                break;
        }
    return myCommand;
}

/// <summary>
/// Imports the data.
/// </summary>
/// <returns></returns>
protected bool ImportData()
{
    bool results = false;
    using (SqlConnection connection = new SqlConnection("Initial Catalog=test;Server=RAGHAV;uid=sa;password=SQL123;"))
    {
        if (ViewState["filePath"] != null)
        {
            using (SqlCommand command = new SqlCommand(GetSqlCommand(), connection))
            {
                SqlParameter paramDataSource = new SqlParameter("@DATASOURCE", SqlDbType.NVarChar, 200)
                {
                    Value =
                        ViewState["filePath"]
                };
                command.Parameters.Add(paramDataSource);

                SqlParameter paramHasHeader = new SqlParameter("@HASHEADER", SqlDbType.NVarChar, 200) { Value = optHasHeader.Checked ? "Yes" : "No" };
                command.Parameters.Add(paramHasHeader);

                SqlParameter paramSheet = new SqlParameter("@SHEET", SqlDbType.NVarChar, 200) { Value = ddlNoOfSheets.SelectedItem.Text };
                command.Parameters.Add(paramSheet);

                connection.Open();
                command.CommandType = CommandType.StoredProcedure;
                results = (command.ExecuteNonQuery() > 0) ? true : false;
            }
        }
    }
}

```

```
        return results;
    }
}

/// <summary>
/// Handles the Click event of the btnImportData control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.EventArgs"/> instance containing the event data.</param>
protected void btnImportData_Click(object sender, EventArgs e)
{
    if (ddlNoOfSheets.SelectedValue=="0")
    {
        lblFileUploadedStatus.Text = "Please select the sheet";
        divAfterFileImport.Visible = false;
    }
    else
    {
        lblImportData.Text = ImportData()
            ? "File has been imported to your database"
            : "There was an error in processing the data";
        divAfterFileUpload.Visible = false;
        divAfterFileImport.Visible = true;
    }
}
}
}
```

I have put some test files in ExcelFile folder situated at the root of the project.

1. DemoFile-97-2003-Format-WithHeader.xls
2. DemoFile-97-2003-Format-WithoutHeader.xls
3. DemoFile-2007-Format-WithHeader.xlsx
4. DemoFile-2007-Format-WithoutHeader.xlsx

You can test for all the cases.

Here is the attached [HowToImportExcelData](#) project rar file.

Do let me know your feedback, comments.

tags : Excel, L, P.NET