# Connecting an ASP.NET-form to a database

Connecting an ASP.NET form created with ExcelEverywhere to a database is very easy. We will do it in 3 steps:

1. Calculate and save the form contents into a database.
2. Retrieve previous entered data from the database, show it in the form and let the user edit it and recalculated and save it again
3. Show all submitted entries so that we can click on them to edit them.

Step 1 can be implemented in several other ways. You do not even need an ASP.NET-page for that, for example ExcelEverywhere for HTML + our advanced service or an external tool like Frontpage extension can be used for that.

We are using Visual Studio 2003. If you do not have it, there are free or cheap alternative IDEs out there: Visual Web Developer 2005 Express Edition Beta, and ASP.NET Web Matrix. Actually, you do not even need an IDE, a text editor like notepad plus DOT.NET version 1.1 is enough.

You can download all files in this example by clicking here.
You need to use ExcelEverywhere for ASP&ASP.NET version 3.2.3 or later for this sample. 3.2.2 will not work.

## Part 1: Saving a ASP.NET-form into a database

Part 1 is to save the entered data into a database. In part 1, the saved data cannot be accessed from the website. You have to access the database file directly to view the submitted data.

## The starting point: the Excel spreadsheet

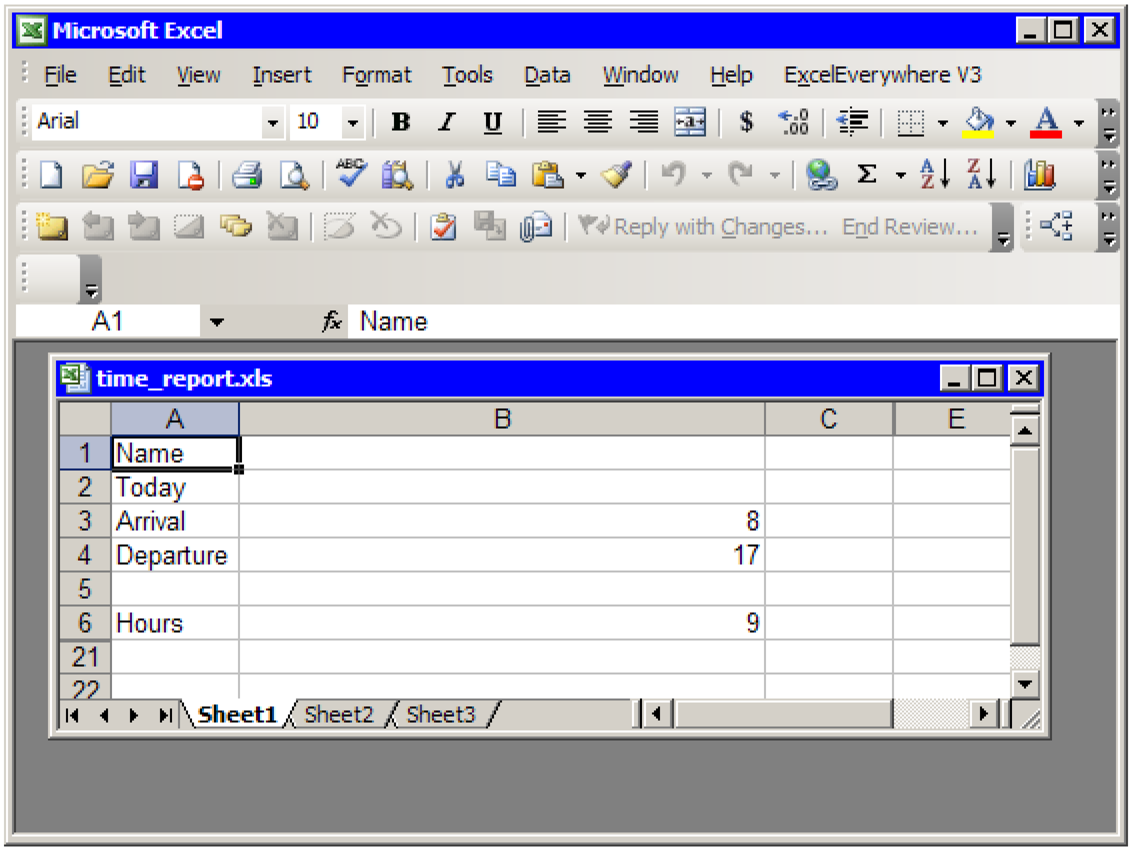
*Todo: create a spreadsheet and name the input and output cells we want to store in the database.*
We started by creating a simple spreadsheet in Excel. It is a very simple and naive time reporting form. The user has to enter name, date, arrival time to work, and departure time. The spreadsheet calculates the number of hours at work.
There are 4 input fields:

1. Name
2. Today
3. Arrival
4. Departure

There is 2 output fields:

1. Hours, the number of hours worked
2. An error message fields which asks the user to arrive before he leaves.

There is a 5th input field called serialno, which cab be ignored now. It is used in order to handle editing of already entered data.


Microsoft Excel
File Edit View Insert Format Tools Data Window Help ExcelEverywhere V3
Arial 10
Reply with Changes... End Review...
A1
Name
time_report.xls
Sheet1 Sheet2 Sheet3

|   | A | B | C | E |
|---|---|---|---|---|
| 1 | Name | | | |
| 2 | Today | | | |
| 3 | Arrival | 8 | | |
| 4 | Departure | 17 | | |
| 5 | | | | |
| 6 | Hours | 9 | | |
| 21 | | | | |
| 22 | | | | |

We have called the spreadsheet time_report.xls and also set the title under File-Properties to 'Report working hours'.
The important is that we have named the cells

- B1: name
- B2: today2
- B3: arrival
- B4: departure
- B6: hours
- C1: serialno

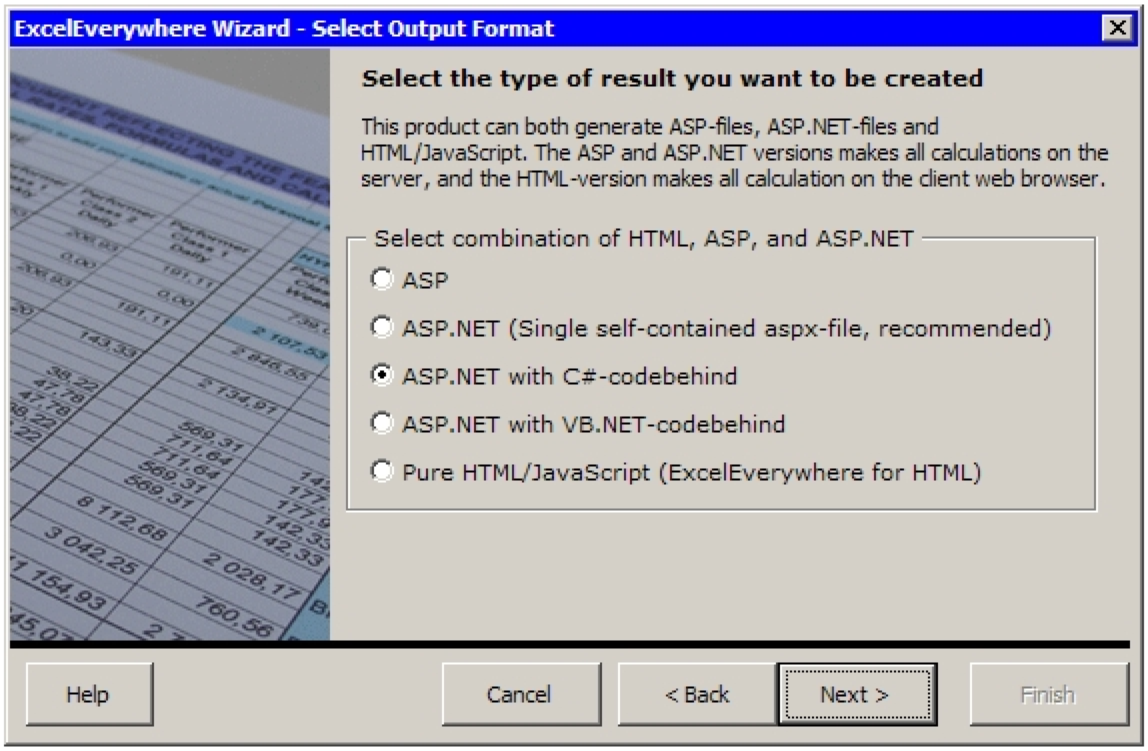
Naming cells is done by placing the cursor in the cell, and writing the name into the small textbox at the top left corner.
Originally, I called the date field today. However, today, seems to be a reserved name in ASP.NET, so I renamed it to today2. Today seems to be a reserved name, I had to rename it to today2. When you remove names in Excel, delete the old name and create a new one. You have to remove the old one, since otherwise ExcelEverywhere might use it.

Make sure you both hide the column with 8-22 and the rows, otherwise you will get blank white space at the bottom of the form.

## Generate the ASP.NET-web page

*Todo: Create an ASP.NET-page with C#-codebehind.*

Start ExcelEverywhere by select Convert from the ExcelEverywhere menu in Excel.
We will select to create an ASP.NET page with C#-codebehind.
All our adaptions are placed in the C#-codebehind file. This means that we will be able to add formatting, change fonts and colors to the sheet, insert new rows at the top, and regenerate our ASP.NET-page. The layout will change, but our integration into the database will remain unchanged!


ExcelEverywhere Wizard - Select Output Format

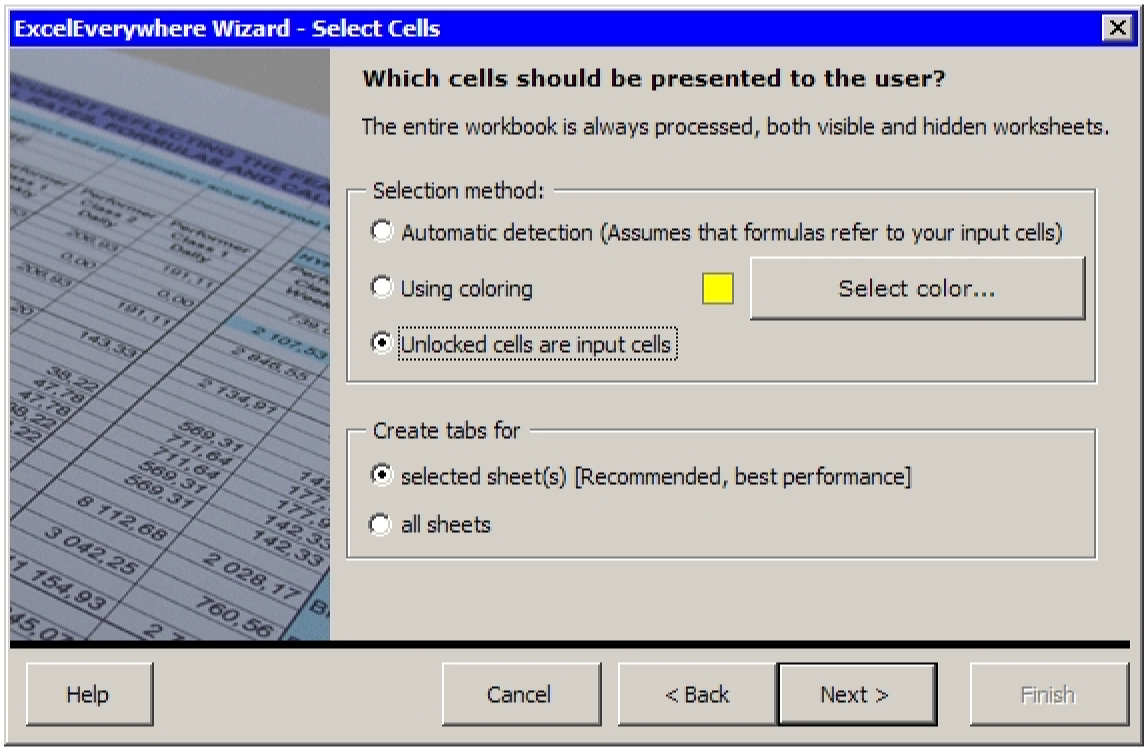
**Select the type of result you want to be created**

This product can both generate ASP-files, ASP.NET-files and HTML/JavaScript. The ASP and ASP.NET versions makes all calculations on the server, and the HTML-version makes all calculation on the client web browser.

Select combination of HTML, ASP, and ASP.NET

- ASP
- ASP.NET (Single self-contained aspx-file, recommended)
- [✓ ASP.NET with C#-codebehind]
- ASP.NET with VB.NET-codebehind
- Pure HTML/JavaScript (ExcelEverywhere for HTML)

All cells in the spreadsheet are locked by default, and we have unlocked the 4 input cells so that ExcelEverywhere can identify them.



ExcelEverywhere Wizard - Select Cells

**Which cells should be presented to the user?**

The entire workbook is always processed, both visible and hidden worksheets.

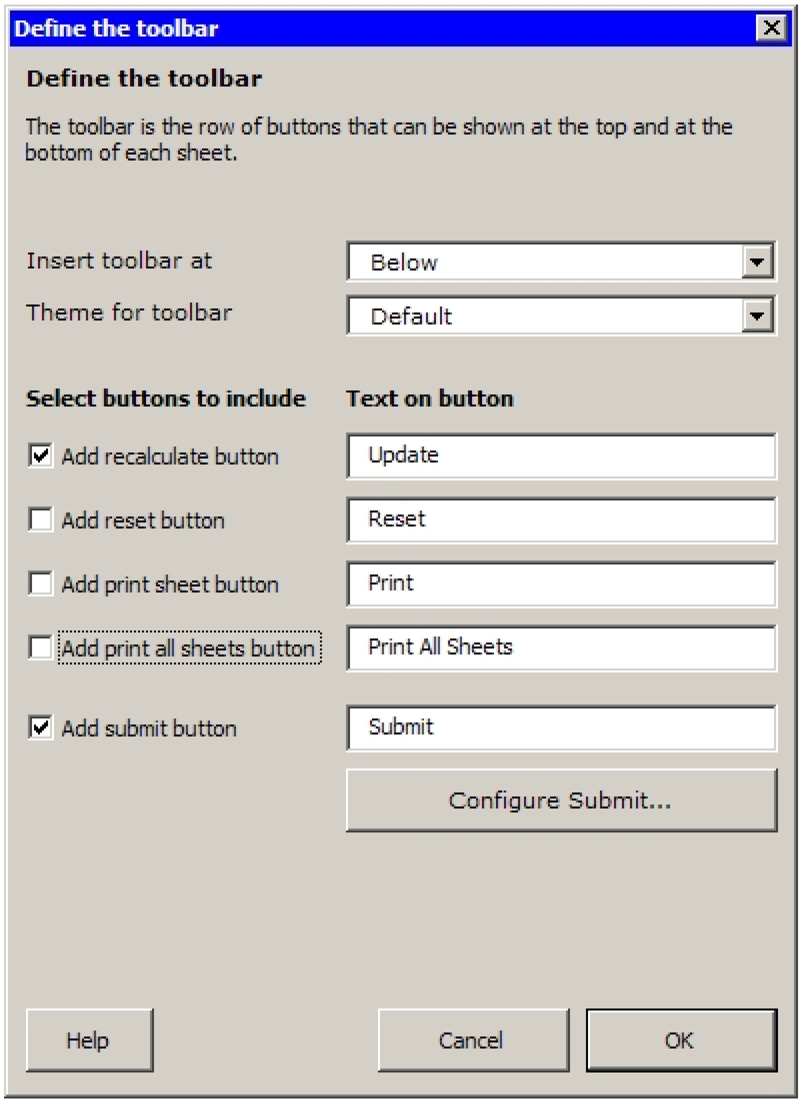
Selection method:

- ( ) Automatic detection (Assumes that formulas refer to your input cells)
- ( ) Using coloring 🟨 Select color...
- (●) Unlocked cells are input cells

Create tabs for

- (●) selected sheet(s) [Recommended, best performance]
- ( ) all sheets

We only keep the update button, which is the button that triggers a recalculation, and the submit button. The submit button will trigger the saving into the database.



Define the toolbar

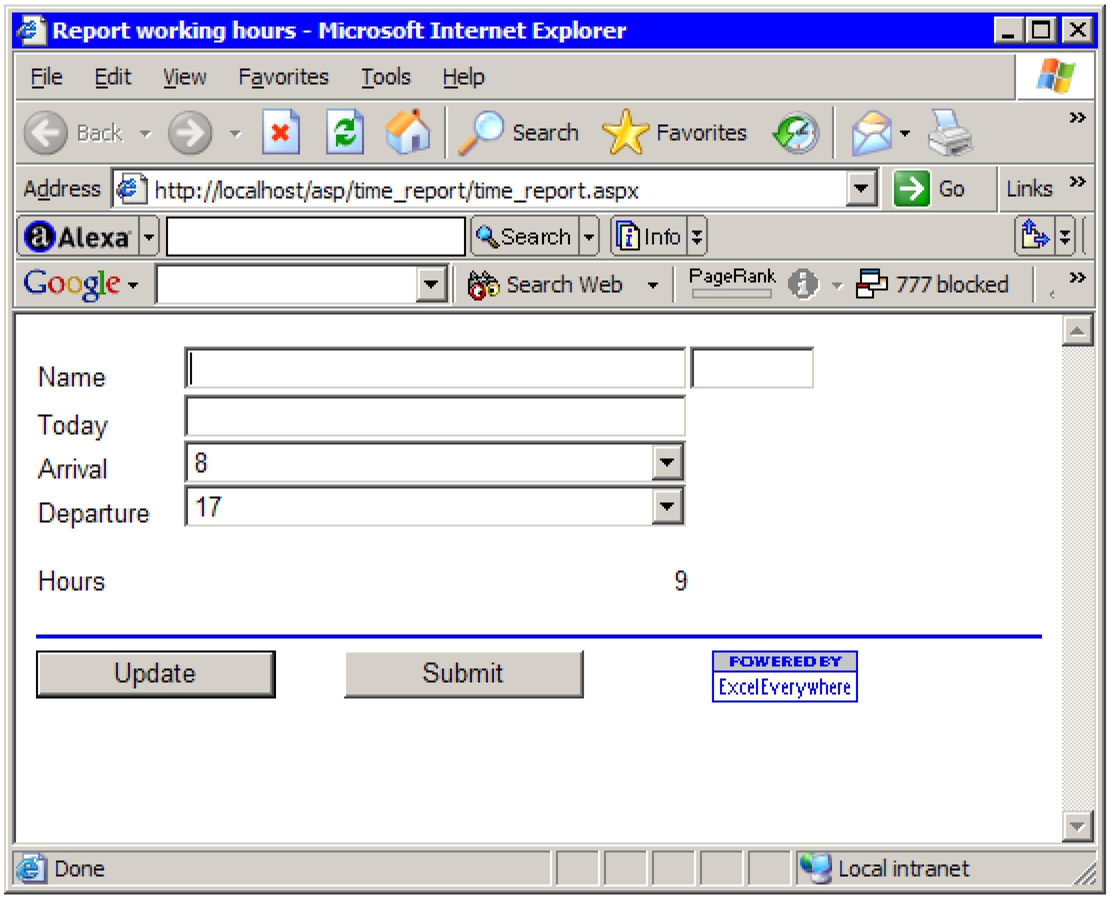
**Define the toolbar**

The toolbar is the row of buttons that can be shown at the top and at the bottom of each sheet.

| | |
|---|---|
| Insert toolbar at | Below |
| Theme for toolbar | Default |

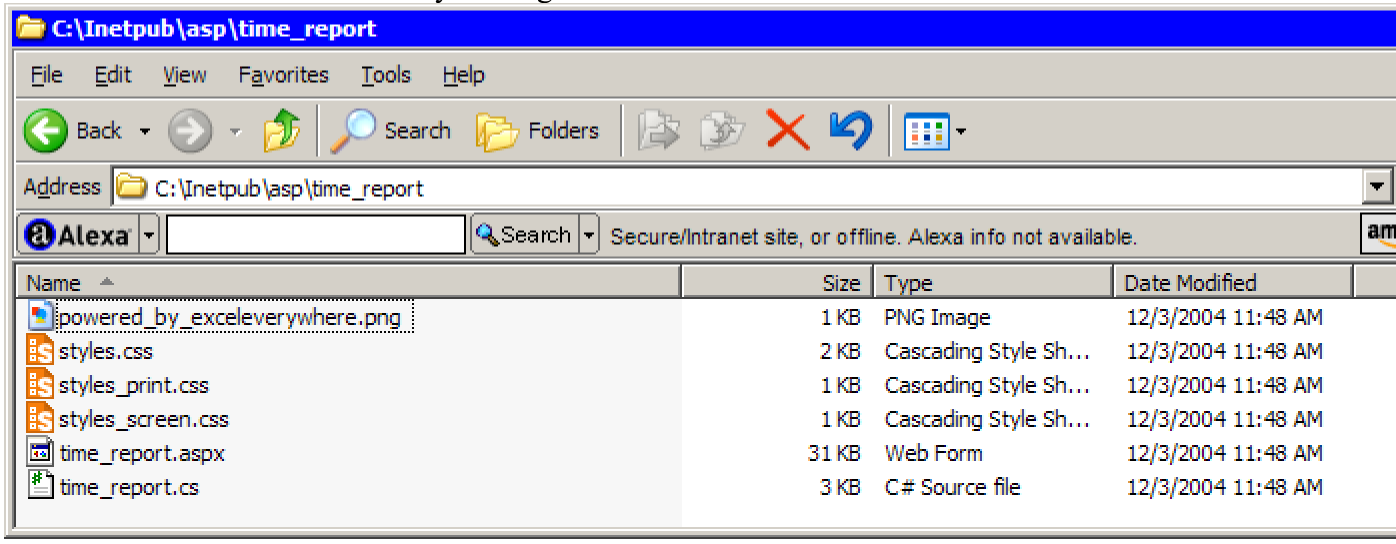
| **Select buttons to include** | **Text on button** |
|---|---|
| ☑ Add recalculate button | Update |
| ☐ Add reset button | Reset |
| ☐ Add print sheet button | Print |
| ☐ Add print all sheets button | Print All Sheets |
| ☑ Add submit button | Submit |

We can test the form directly. Pressing Update will recalculate the values. Submit is not yet implemented and will not do anything.


Report working hours - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/asp/time_report/time_report.aspx Go Links
Alexa Search Info
Google Search Web PageRank 777 blocked
Update
Submit
POWERED BY ExcelEverywhere
Done
Local intranet

| | |
|---|---|
| Name | |
| Today | |
| Arrival | 8 |
| Departure | 17 |
| Hours | 9 |

These are the files that ExcelEverywhere generated:


C:\Inetpub\asp\time_report
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Inetpub\asp\time_report
Alexa Search
Secure/Intranet site, or offline. Alexa info not available.

| Name | Size | Type | Date Modified |
|---|---|---|---|
| powered_by_exceleverywhere.png | 1 KB | PNG Image | 12/3/2004 11:48 AM |
| styles.css | 2 KB | Cascading Style Sh... | 12/3/2004 11:48 AM |
| styles_print.css | 1 KB | Cascading Style Sh... | 12/3/2004 11:48 AM |
| styles_screen.css | 1 KB | Cascading Style Sh... | 12/3/2004 11:48 AM |
| time_report.aspx | 31 KB | Web Form | 12/3/2004 11:48 AM |
| time_report.cs | 3 KB | C# Source file | 12/3/2004 11:48 AM |

## Creating the database

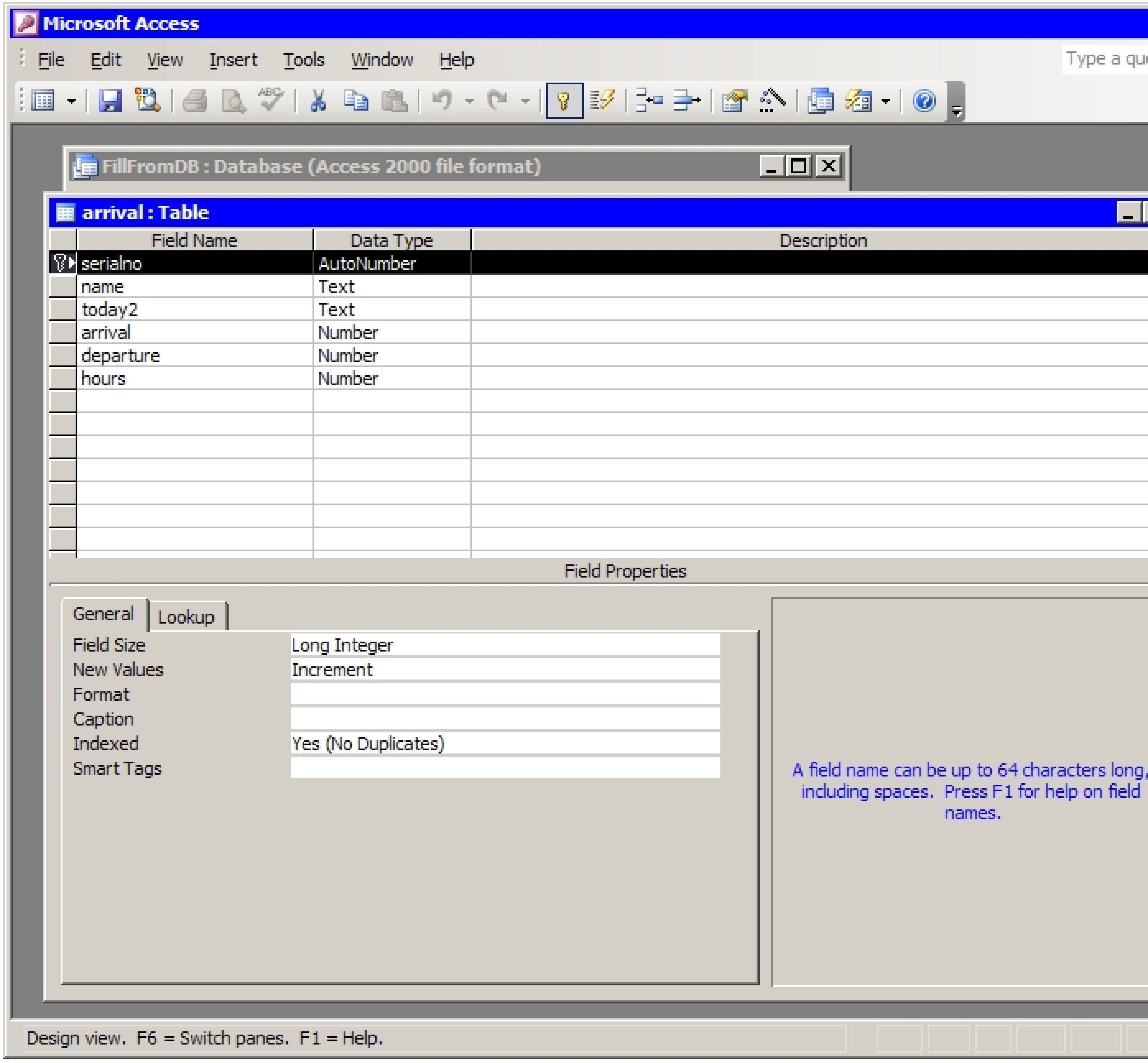
*Todo: We need to create a database where the values can be stored.*

The database will only contain one single table called arrival. Each submit will add one row to that table.
Let us create a database with one single table where we can store the columns. The columns of the table arrival are

- serialno: long integer, autoincrement
- name: string 50
- today2: string 20 (we could have used a date format)
- arrival: number (single)
- departure: number (single)
- hours: number (single)

where serialno is the primary key. In this first part, serialno will have no purpose.We call the table arrival

We start Access. You can skip this part if you do not have access to MS Access, you can use the empty database in the zip-file.
Drivers for Access is included in Windows.


Microsoft Access
File Edit View Insert Tools Window Help
Type a qu
FillFromDB : Database (Access 2000 file format)
arrival : Table
Field Properties
General Lookup

| Field Name | Data Type | Description |
|---|---|---|
| serialno | AutoNumber | |
| name | Text | |
| today2 | Text | |
| arrival | Number | |
| departure | Number | |
| hours | Number | |

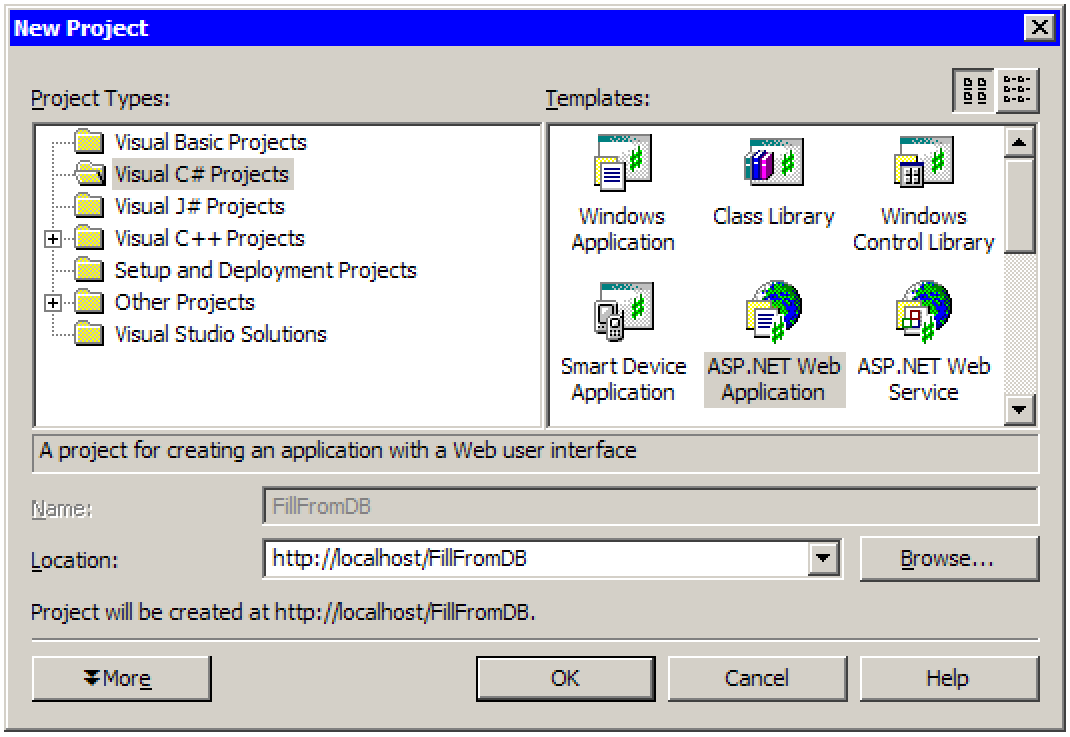
| | |
|---|---|
| Field Size | Long Integer |
| New Values | Increment |
| Format | |
| Caption | |
| Indexed | Yes (No Duplicates) |
| Smart Tags | |

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

## Creating the ASP.NET-application

*Todo: Create an ASP.NET application and import the ASP.NET-page we created with ExcelEverywhere.*
Start Visual Studio 2003 and create a new C# ASP.NET application project called FillFromDB.


New Project

Project Types:

- Visual Basic Projects
- Visual C# Projects
- Visual J# Projects
- Visual C++ Projects
- Setup and Deployment Projects
- Other Projects
- Visual Studio Solutions

Templates:

- Windows Application
- Class Library
- Windows Control Library
- Smart Device Application
- ASP.NET Web Application
- ASP.NET Web Service

A project for creating an application with a Web user interface

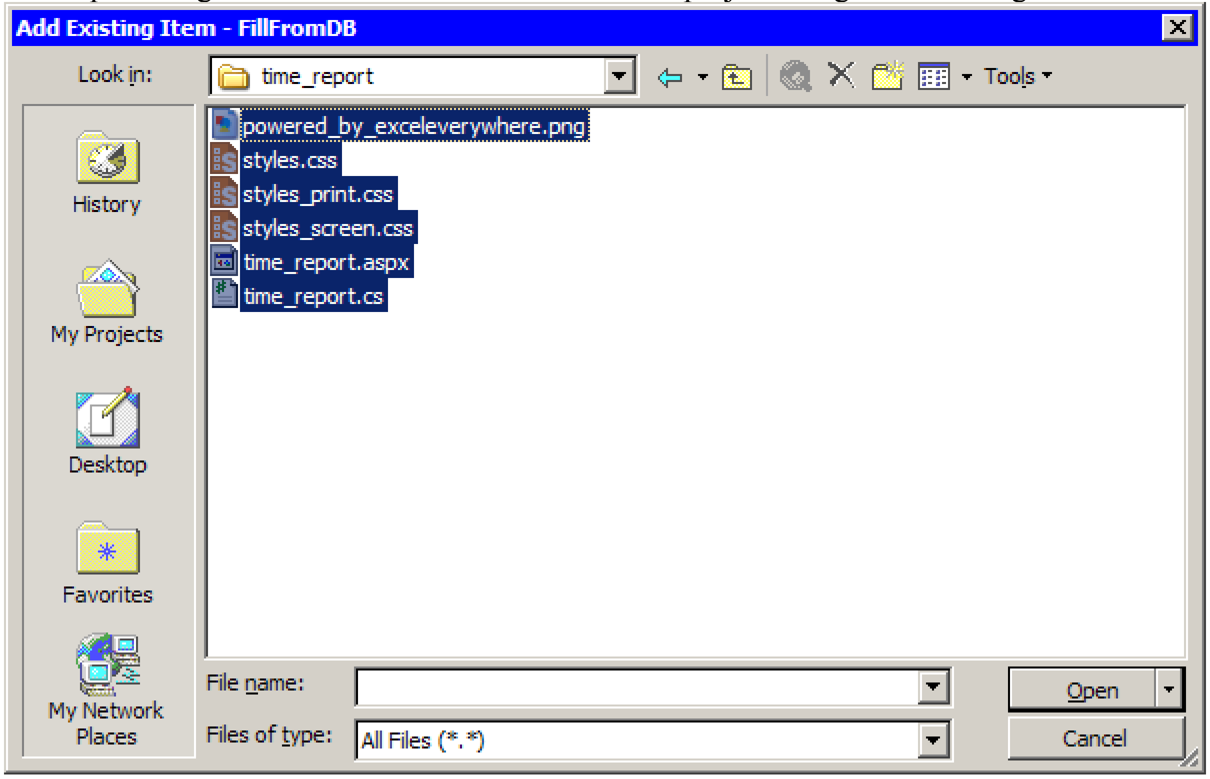
Name: FillFromDB

Location: http://localhost/FillFromDB

Project will be created at http://localhost/FillFromDB.

## Import the ASP.NET-page into Visual Studio 2003

We import the generated files into our Visual Studio project using Add Existing Item.


Add Existing Item - FillFromDB

Look in: time_report

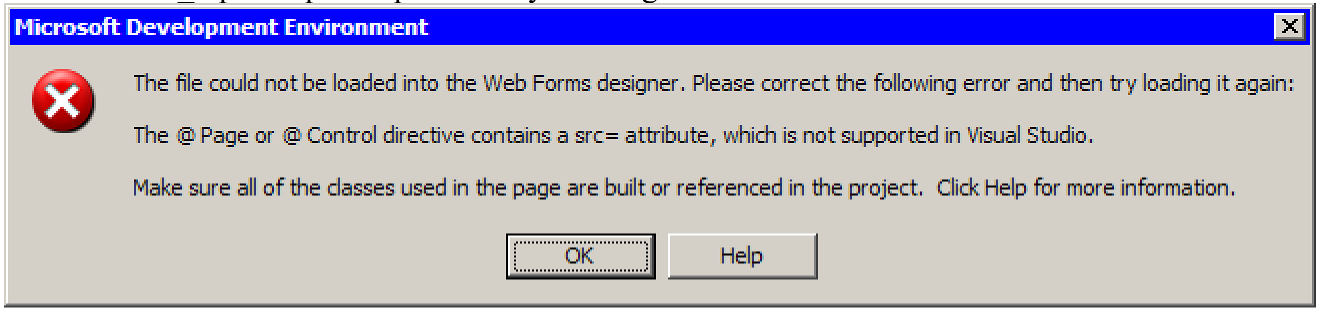
- powered_by_exceleverywhere.png
- styles.css
- styles_print.css
- styles_screen.css
- time_report.aspx
- time_report.cs

File name:

Files of type: All Files (*.*)

## Delete the Src= tag

Click on time_report.aspx to open it and you will get



Microsoft Development Environment

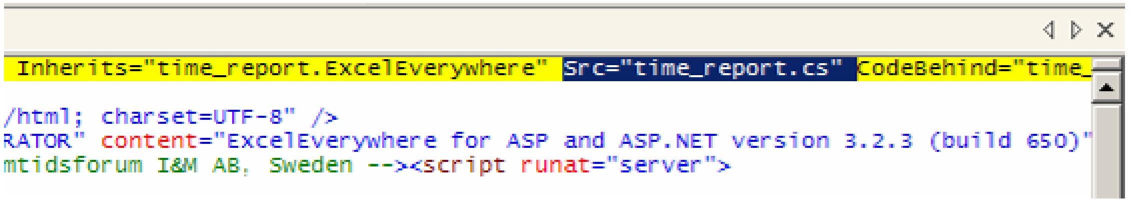
The file could not be loaded into the Web Forms designer. Please correct the following error and then try loading it again:

The @ Page or @ Control directive contains a src= attribute, which is not supported in Visual Studio.

Make sure all of the classes used in the page are built or referenced in the project. Click Help for more information.

You have to delete the SRC-attribute. Visual Studio uses CodeBehind instead. A little article describing the difference between Src and CodeBehind:
http://www.dotnetcoders.com/web/Articles/ShowArticle.aspx?ar ticle=22

Delete Src="time_report.cs"

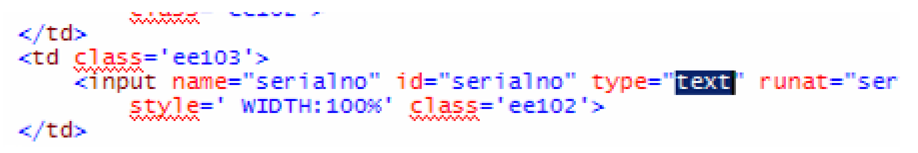


```
 Inherits="time_report.ExcelEverywhere" Src="time_report.cs" CodeBehind="time_

/html; charset=UTF-8" />
RATOR" content="ExcelEverywhere for ASP and ASP.NET version 3.2.3 (build 650)"
mtidsforum I&M AB, Sweden --><script runat="server">
```

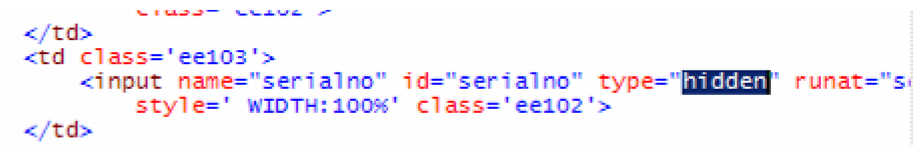
Save and close, then reopen the time_report.aspx.

## Hide internal fields like serialno

We hide the serialno field by opening time_report.aspx and select the HTML-view. Look for
id="serialno" type="text"
and replace it by
id="serialno" type="hidden"



```
 </td>
 <td class='ee103'>
     <input name="serialno" id="serialno" type="text" runat="ser
         style=' WIDTH:100%' class='ee102'>
 </td>
```



```
 </td>
 <td class='ee103'>
     <input name="serialno" id="serialno" type="hidden" runat="s
         style=' WIDTH:100%' class='ee102'>
 </td>
```

We also need to update the codebehind file, the type has changed from HtmlInputText to HtmlInputHidden.

```
        protected System.Web.UI.HtmlControls.HtmlInputText serialno;
```

into

```
        protected System.Web.UI.HtmlControls.HtmlInputHidden serialno;
```

## The initial codebehind file before integration

The generate time_report.cs. Improve the indentation by selecting all and then Edit-Advanced-Format Selection

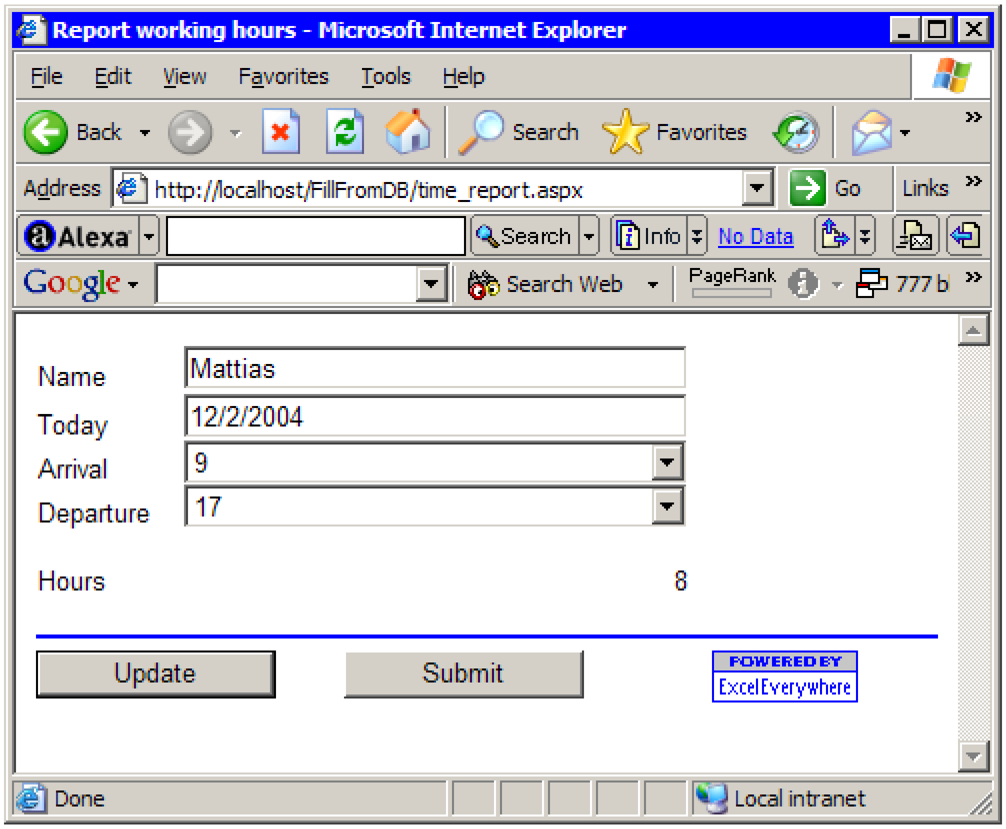
```
// This file will not be overwritten!
//
// This is the recommended location for backend integrations

using System;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace time_report
{

    public abstract class ExcelEverywhere : System.Web.UI.Page
    {
        public abstract void initArrays();
        public abstract void setReset();
        public abstract void fromForm();
        public abstract void docalc();
        public abstract void toForm();
        public abstract Object getObject();

        /* All the ASPX-fields as of when this file was first generated.
       The list is not updated, since this file is not overwritten.
       In order to get the new list. Delete this file. */

        // protected HtmlInputText name;
        // protected HtmlInputText today2;
        // protected HtmlSelect arrival;
        // protected HtmlSelect departure;
        // protected Label p1B5;
        // protected Label hours;


        protected void server_recalc(object objSender, System.EventArgs objArgs)
        {
            // no code here since Page_load handles the recalculation
        }

        protected string panel_to_show = "panel1";

        protected void Page_Load(object objSender, System.EventArgs objArgs)
        {
            initArrays();
            setReset();


            // TIP: Add code here to set attributes of getObject()
            // to override default values
            if (IsPostBack)
            {
                panel_to_show = Request.Params["xl_sheet_no"];
                if (Request.Params["xl_update_top"] != null ||
Request.Params["xl_update_bottom"] != null || Request.Params["xl_submit_top"] != null ||
Request.Params["xl_submit_bottom"] != null)
                {
                    // it wasn't a reset
                    fromForm();

                    // TIP: Add code here to read attributes from getObject()
                    // if you want to save the values entered by the user
                }
            };
            docalc();

            if (Request.Params["xl_submit_top"] != null ||
Request.Params["xl_submit_bottom"] != null)
            {
                // submit was pressed: save and redirect to confirmation page

                // TIP: Add code here to read attributes from getObject()
                // if you want to save the values entered by the user
                // and the calculated values.
            }
            toForm();
        }
    }
}
```

If not VS2003 has created these attributes automatically, remove the comments before the field names in the form. Skip p1B5, which is the error message. For me VS2003 created them all except for the today2-field.

```
        protected HtmlInputText name;
        protected HtmlInputText today2;
        protected HtmlSelect arrival;
        protected HtmlSelect departure;
        // protected Label p1B5;
        protected Label hours;
        protected HtmlInputHidden serialno;
```

## Test the imported ASPNET-page

Set time_report.aspx as start page and compile and run. Enter some data and press Update the hours should be recalculed.

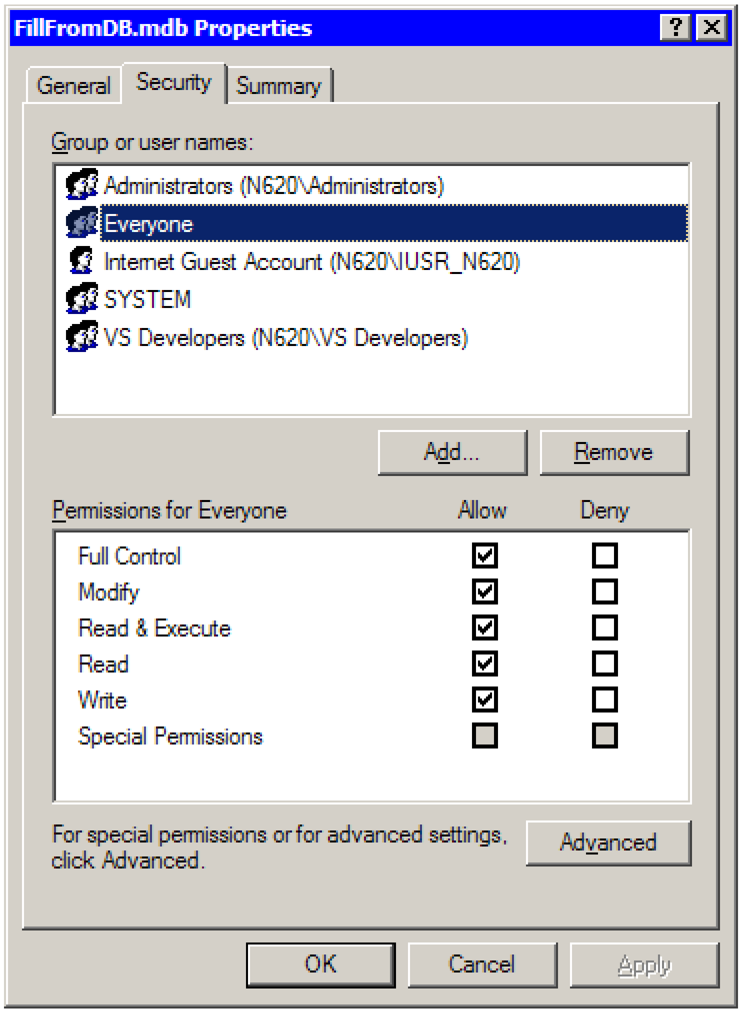

Report working hours - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/FillFromDB/time_report.aspx Go Links
Alexa Search Info No Data
Google Search Web PageRank 777 b
Update
Submit
POWERED BY ExcelEverywhere
Done
Local intranet

|  |  |
|---|---|
| Name | Mattias |
| Today | 12/2/2004 |
| Arrival | 9 |
| Departure | 17 |
| Hours | 8 |

So now the ASP.NET-page has been accepted by Visual Studio 2003 and we can start with the integration. If you want to improve the formatting of the web page, update the time_report.xls spreadsheet, and regenerate the web page. The only two files that you should update in your VS2003-project is time_report.aspx and styles.css. The other files should be kept. You will have to remove the src=”time_report.cs” and hide serialno each time.

## Include the database into the project

Place or import the database file, i.e. fillfromdb.mdb in the bin directory. Right-click on the file, select properties and set the security of the file. I let Everyone has Full Control, which actually is a bit dangerous. I should have given Full Control to the ASPNET-account used by IIS.


FillFromDB.mdb Properties

General | Security | Summary

Group or user names:

- Administrators (N620\Administrators)
- Everyone
- Internet Guest Account (N620\IUSR_N620)
- SYSTEM
- VS Developers (N620\VS Developers)

| Permissions for Everyone | Allow | Deny |
|---|---|---|
| Full Control | ☑ | ☐ |
| Modify | ☑ | ☐ |
| Read & Execute | ☑ | ☐ |
| Read | ☑ | ☐ |
| Write | ☑ | ☐ |
| Special Permissions | ☐ | ☐ |

For special permissions or for advanced settings, click Advanced.

## Save submitted data to database

Todo: Add code that opens that database, reads the values from the form, saves them into the database and then closes database.

Find the comment `// submit was pressed: save and redirect to confirmation page` In the code above. That is where we will insert our database code.

We have to adjust the code slightly. Move toForm() directly after docalc() since we need the calculated values during save. The new code looks like:

```
            docalc();

            toForm();  // update form since we read output values below

            if (Request.Params["xl_submit_top"] != null ||
Request.Params["xl_submit_bottom"] != null)
```

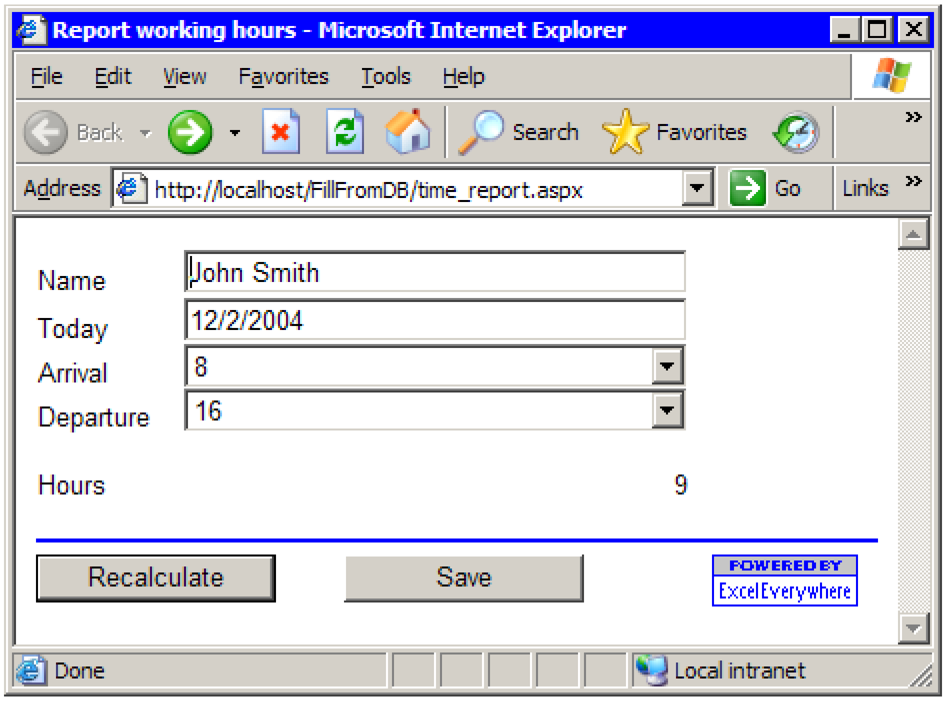
The value of a form fields like name are accessed with name.Value. The SQL-statement that insert the row into the database looks like

```
"INSERT INTO arrival(arrival, departure, hours, name, today2) VALUES (" + arrival.Value +
"," + departure.Value +, " + hours.Value +, " + name.Value +, " + today2.Value +)"
```

However, you should never build SQL-statements in this way. It opens up the application for SQL injection attacks, which enables malicious user to access and delete your database.
Instead, you should always use SQL parameters, preferable typed parameters. So the simple line above becomes

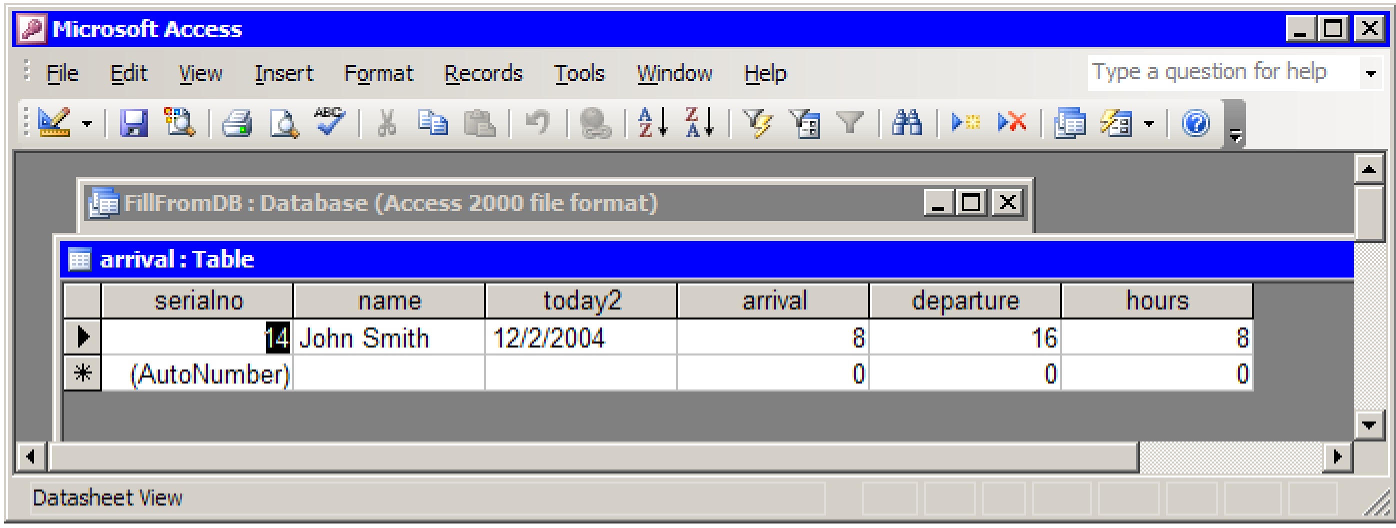
```
                OleDbConnection oleDbConnection1 = new
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=c:\Inetpub\wwwroot\FillFromDB\bin\FillFromDB.mdb");
                try
                {
                    oleDbConnection1.Open();

                        OleDbCommand oleDbInsertCommand1 = new OleDbCommand();
                        oleDbInsertCommand1.CommandText = "INSERT INTO arrival(arrival,
departure, hours, name, today2) VALUES (?, ?, ?, ?, ?)";
                        oleDbInsertCommand1.Connection = oleDbConnection1;
                        oleDbInsertCommand1.Parameters.Add(new
System.Data.OleDb.OleDbParameter("arrival", System.Data.OleDb.OleDbType.Single, 0,
"arrival"));
                        oleDbInsertCommand1.Parameters.Add(new
System.Data.OleDb.OleDbParameter("departure", System.Data.OleDb.OleDbType.Single, 0,
"departure"));
                        oleDbInsertCommand1.Parameters.Add(new
System.Data.OleDb.OleDbParameter("hours", System.Data.OleDb.OleDbType.Single, 0,
"hours"));
                        oleDbInsertCommand1.Parameters.Add(new
System.Data.OleDb.OleDbParameter("name", System.Data.OleDb.OleDbType.VarWChar, 50,
"name"));
                        oleDbInsertCommand1.Parameters.Add(new
System.Data.OleDb.OleDbParameter("today2", System.Data.OleDb.OleDbType.VarWChar, 20,
"today2"));


                        oleDbInsertCommand1.Parameters["arrival"].Value =
Convert.ToSingle(arrival.Value);
                        oleDbInsertCommand1.Parameters["departure"].Value =
Convert.ToSingle(departure.Value);
                        oleDbInsertCommand1.Parameters["hours"].Value =
Convert.ToSingle(hours.Text);
                        oleDbInsertCommand1.Parameters["name"].Value = name.Value;
                        oleDbInsertCommand1.Parameters["today2"].Value = today2.Value;

                        try
                        {
                            int val = oleDbInsertCommand1.ExecuteNonQuery();
                            Response.Write("<p>Created</p>");
                        }
                        catch (OleDbException /* exc */)
                        {
                            Response.Write("<p><strong>Failed to insert data into
database</strong></p>");
                        }

                        // we need to reset the form, since we do not know the serialno
                        initArrays();
                        setReset();
                        docalc();
                        toForm();

                }
                finally
                {
                    oleDbConnection1.Close();
                }
```

If we start the application and enters a person


Report working hours - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/FillFromDB/time_report.aspx Go Links
Recalculate
Save
POWERED BY ExcelEverywhere
Done
Local intranet

| | |
|---|---|
| Name | John Smith |
| Today | 12/2/2004 |
| Arrival | 8 |
| Departure | 16 |
| Hours | 9 |

And look at the database from Access:


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
FillFromDB : Database (Access 2000 file format)
arrival : Table
Datasheet View

| serialno | name | today2 | arrival | departure | hours |
|---|---|---|---|---|---|
| 14 | John Smith | 12/2/2004 | 8 | 16 | 8 |
| (AutoNumber) | | | 0 | 0 | 0 |

You can also view the database from inside Visual Studio 2003. Right-click on the Server Explorer, Add connection, select Microsoft Jet 4.0 OLE DB Provider, and select the file MDB-file.
Tip: Close the database in Access after viewing it, otherwise the database might be locked.

## Which database to use: Microsoft Access or Microsoft SQL Server?

If you use Microsoft Access, the code prefix for the classes is OleDb, if you use SqlServer, it is Sql, ie. `OleDbParameter` and `SqlParameter`. If you only need to save data and no advanced querying is needed Access is a good database. Reliabilty for Access has increased during the years, but SqlServer is a better database. Microsoft has a free version of MS Sql Server called MSDE.

## SQL injection

You can read more about SQL injection and why we have to build parameterized SQL-statements here:

- http://www.developer.com/db/article.php/2243461
- http://dotnetjunkies.com/WebLog/richard.dudley/articles/13706.aspx
- http://blogs.wdevs.com/ColinAngusMackay/archive/2004/09/25/652.aspx
- http://weblogs.asp.net/bleroy/archive/2004/08/18/216861.aspx

A general checklist on how to secure ASP.NET-sites is found here:

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/CL_SecuAsp.asp
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/THCMCh10.asp

## Conclusion

Adding code to an ASP.NET-page that stores the contents into a database is easy. By placing the database code into the codebehind file, it will not interfere with the generated code, and you and update formulas and formatting in the spreadsheet later, and still keep you backend integration code intact.
In the next step, we will add more database code that fills the form with data from the database and lets the user update it.