

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.](#)

[You can read the 1st part here.](#)

Part 2: Presenting and updating data in the form

Filling the form with data from the database is almost as easy as saving the data. The only complication is that there must be a way to tell the ASP.NET-form which entry to display. We have chosen the standard solution where each entry has a unique that is used to find the entry.

Design

Each entry in the database as a unique serial number column, called serialno. The database automatically fills in this column.

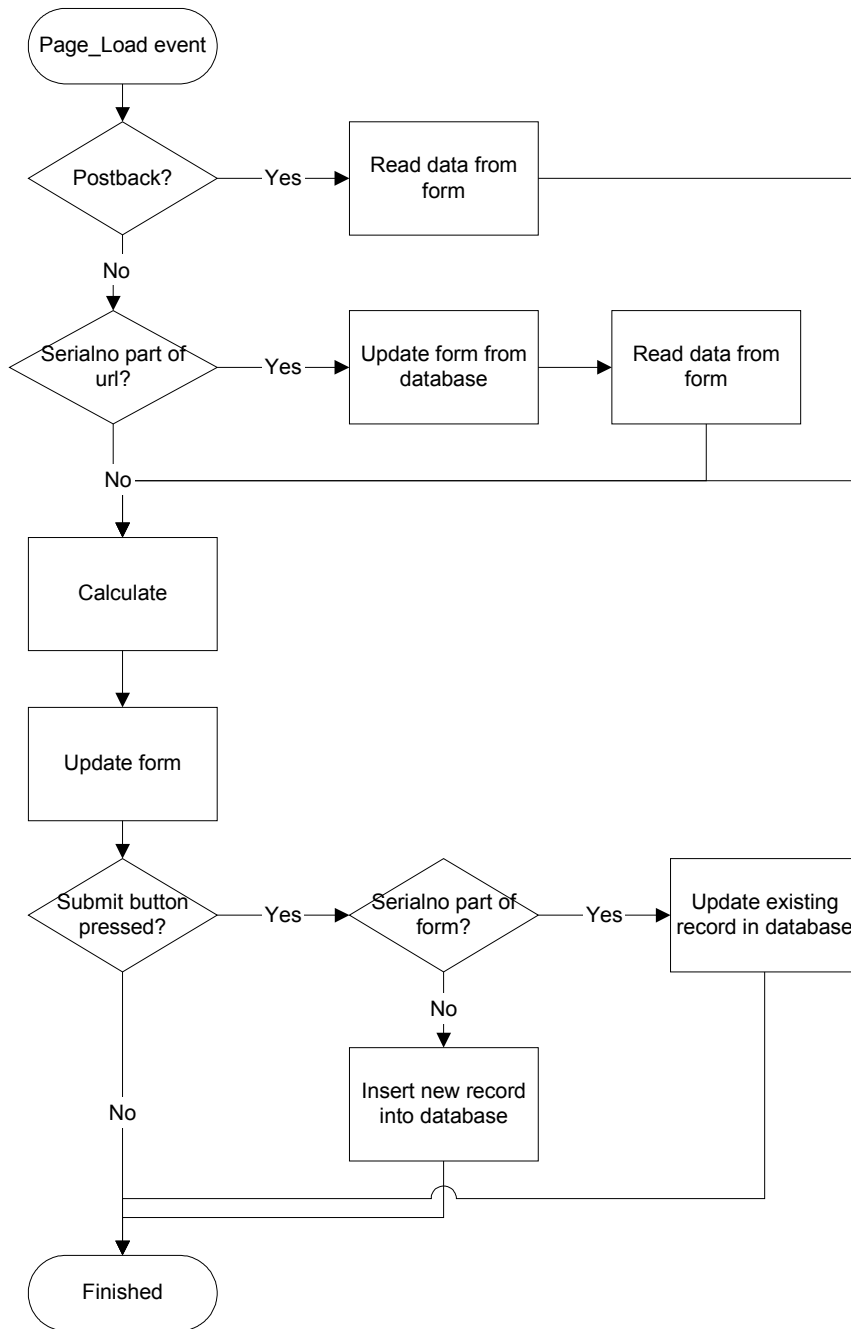
The normal url to our ASP.NET-form is http://localhost/FillFromDB/time_report.aspx.

When the web page is called the first time, it returns an empty form, when the web page is called after that, the calculated fields are automatically recalculated.

In order to handle updating of old data from the database, we add a HTTP Query Parameter to the url, which tells which row in the database to edit. The url with the query parameter will look like http://localhost/FillFromDB/time_report.aspx?serialno=14

If the contents of the form is read from the database, we do not want to add a new record to the database when the user presses Save. In order to remember where the data belongs, we remember the serialno from the url in the form field which also is called serialno. (We could have named the fields differently.)

Whenever the user wants to save the form, we look at the form field serialno. If it is set, then we should update an existing row in the database. If serialno is not set, this is a new row, and we insert the data into the database.



SQL-statement for finding an old time report.

As described in part 1, we should always use parameterized SQL-statements. The structure of the SELECT-statement, which is the one that reads an old entry is
 SELECT arrival, departure, hours, name, serialno, today2 FROM arrival where serialno = ?

SQL-statement for updating an old time report

The structure of the UPDATE statement which replaces the old values of arrival, departure, hours, name and today2 with new values is:
UPDATE arrival SET arrival = ?, departure = ?, hours = ?, name = ?, today2 = ? WHERE (serialno = ?)

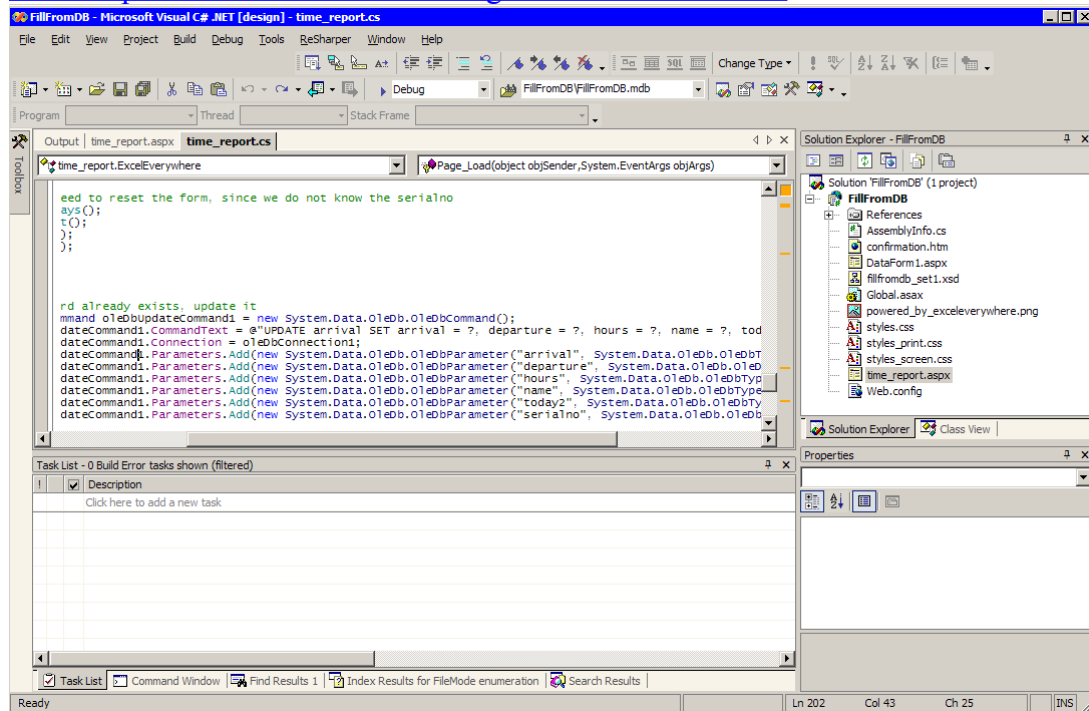
SQL-statement for inserting a new time report

The structure of the INSERT statement hasn't changed. Note that serialno is not mentioned. The database handler will set that column automatically, since it is an auto-increment number.

INSERT INTO arrival(arrival, departure, hours, name, today2) VALUES (?, ?, ?, ?, ?)

The complete SQL-code

[The complete source code is a bit long. Click here to view it.](#)

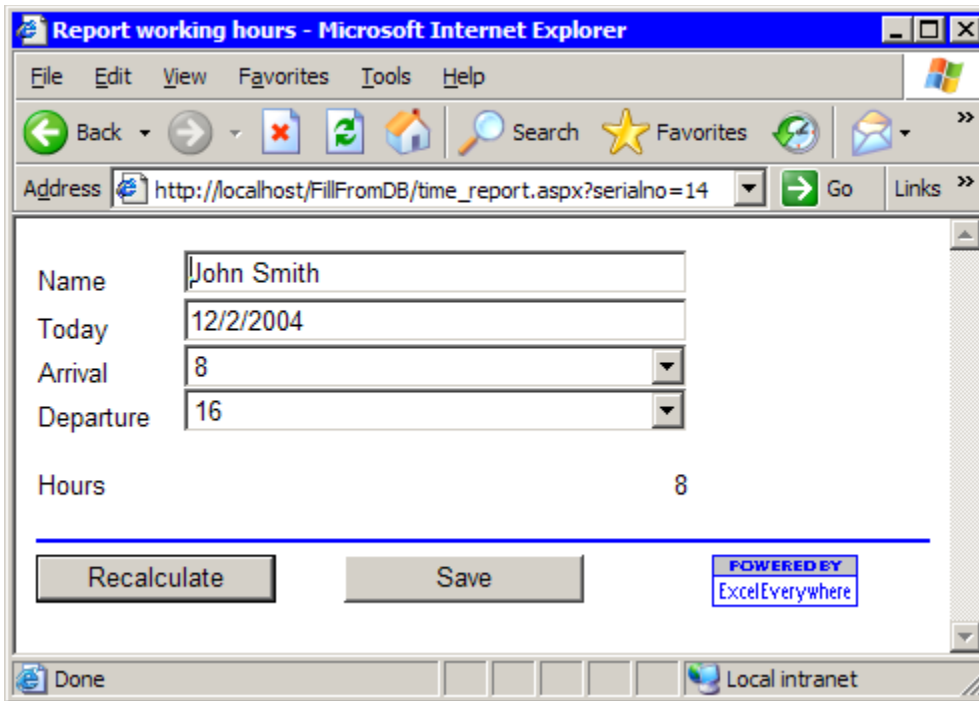


Testing updating an old value

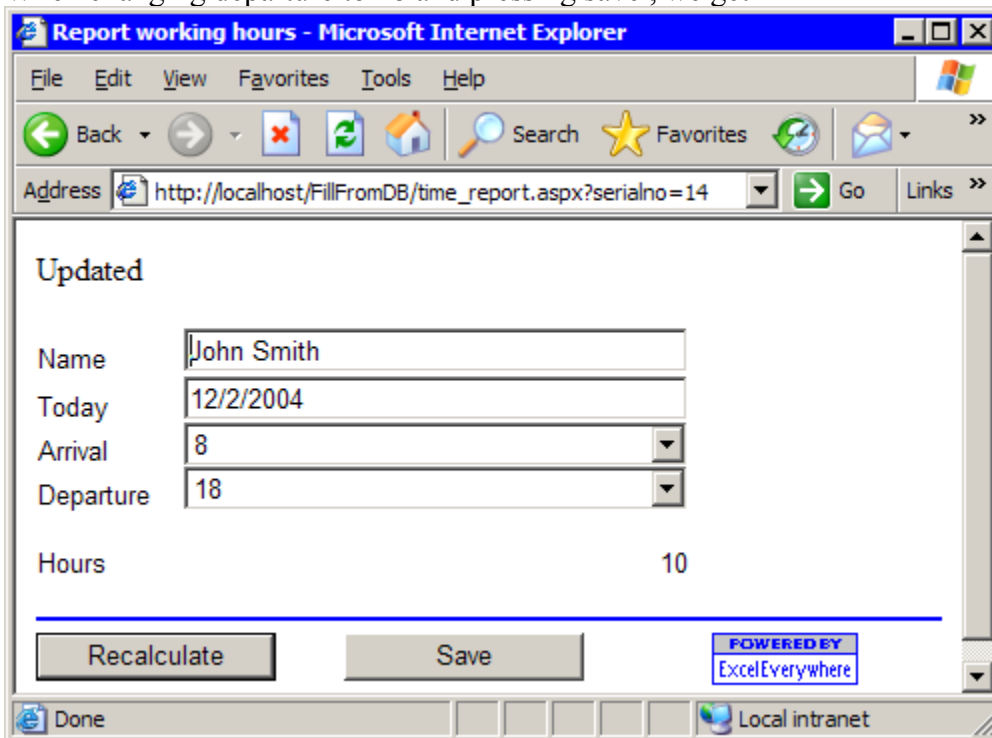
I looked into my database using Microsoft Access and saw that I had an entry with serialno = 14.

The link to edit that row is http://localhost/FillFromDB/time_report.aspx?serialno=14.

Clicking on that URL, the form opens with



When changing departure to 18 and pressing save , we get



Looking into the database, I see that the rows has changed.

Conclusion

We added code that reads an old entry, and code that updates an old entry. Although the code is rather long, mostly depending on the fact that we have to use parameterized SQL-statements to get a secure site, it is simple. In order to adapt this to your form, all you need to use is cut-and-paste and rename the column names.

Accessing old entries with the URL

http://localhost/FillFromDB/time_report.aspx?serialno=14 is not very nice, and in the next part we will use the DataForm-wizard to create a clickable list of the existing entries. Also note that all code has been added to the codebehind file, so we can edit the spreadsheet and regenerate the aspx-file and replace the current version. All we have to do is hide the serialno field and delete the SRC-tag.