2010

CODE ONTIME

COOKBOOK

CreatedBy, CreatedOn, ModifiedBy, ModifiedOn,…

Let's add tracking of user activity in the *Customers* table of *Northwind* database. We'll start by adding four new fields: *CreatedBy*, *CreatedOn*, *ModifiedBy*, and *ModifiedOn*. This can be done in *SQL Server Management Studio*.

	Column Name	Data Type	Allow Nulls
🔑	CustomerID	nchar(5)	☐
	CompanyName	nvarchar(40)	☐
	ContactName	nvarchar(30)	☑
	ContactTitle	nvarchar(30)	☑
	Address	nvarchar(60)	☑
	City	nvarchar(15)	☑
	Region	nvarchar(15)	☑
	PostalCode	nvarchar(10)	☑
	Country	nvarchar(15)	☑
	Phone	nvarchar(24)	☑
	Fax	nvarchar(24)	☑
	CreatedBy	nvarchar(50)	☑
	CreatedOn	datetime	☑
	ModifiedBy	nvarchar(50)	☑
	ModifiedOn	datetime	☑
			☐

Customers *

Now let's create a *Web Site Factory* application that will automatically populate these fields with relevant data. Give it the name of "TrackingChanges".

New Project

Vendor: Code OnTime LLC
http://www.codeontime.com

Web Site Factory

Create an ASP.NET/AJAX web site with navigation system, role-based security, membership manager, built-in data export and reporting, adaptive filtering, advanced search options, and much more. The generated project is managed via the file system folder structure.

Review a complete feature list of this project. Try a live demo now.

Name:

TrackingChanges

Connect to your server and select the *Northwind* database. Use the *Object Selector* to exclude all other tables from the project except the *Customers* table.

Database Connection

A valid connection string compatible with the selected data provider is required to generate the project. Data controller descriptors are automatically built from your database and are easy to maintain with the provided XML schemas.

Data Provider:

.Net Framework Provider for SqlServer (System.Data.SqlClient) ▼

Connection String:

Data Source=db;Initial Catalog=Northwind;Integrated Security=True;

▫ Click here if you need assistance to build the connection string.
▫ Select the database tables and views included in the project.

The following database objects are included in this project:
● dbo.Customers

Enable reporting for the application.

Enable *ASP.NET Membership*. If necessary, you can connect to a standalone membership database for the application.

Continue with the project wizard and generate the application. When finished, a web page will appear with the fresh application. There are only three pages available, *Home*, *Customers*, and *Membership*. Navigate to *Customers*. You can see that the new fields are not visible in grid view, but can be viewed and edited from detail view.

Our goal here is to make the fields non-editable in grid and detail view, and not displayed in *New Customers* page. Switch back to the *Generator*, click on the project name, and press the *Design* button. From the list of *All Controllers*, select the *Customers* controller. Switch to *Fields* tab, and select the first field that was made, *CreatedBy*. Edit the field, and change the *Code Default* field to "Context.User.Identity.Name". This will insert the name of the user creating the record.

Save this field, and click on the next field, *CreatedOn*. In the *Code Default* field, insert "DateTime.Now". This will insert the date that the field was created on. Scroll down, and in the *Data Format String* field, type in "{0:G}" or "G". This will show the date and time, up to the seconds. If lowercase "g" is used, seconds will not be displayed.

Now, save and move on to the *ModifiedBy* field. As this field needs to be updated every time the record is modified, we will use the *Code Value* field. Type in "Context.User.Identity.Name", and save.


Next, go to the *ModifiedOn* field. In the *Code Value* property, type in "DateTime.Now", and in the *Data Format String* property, type in "{0:G}" or "G". These expressions work for both *Visual Basic.NET* and *C#*.


Next, we'll modify the bindings of each field to the respective data views. Select *CreatedBy*, and switch to the *Data Fields* tab. Edit the first binding so that the field is hidden in *createForm1*, and change *Text Mode* to "Static" for *editForm1*.

Save, and go back to the list of fields. Select the *CreatedOn* field, and switch to the *Data Fields* tab. Perform the same changes as with *CreatedOn*.
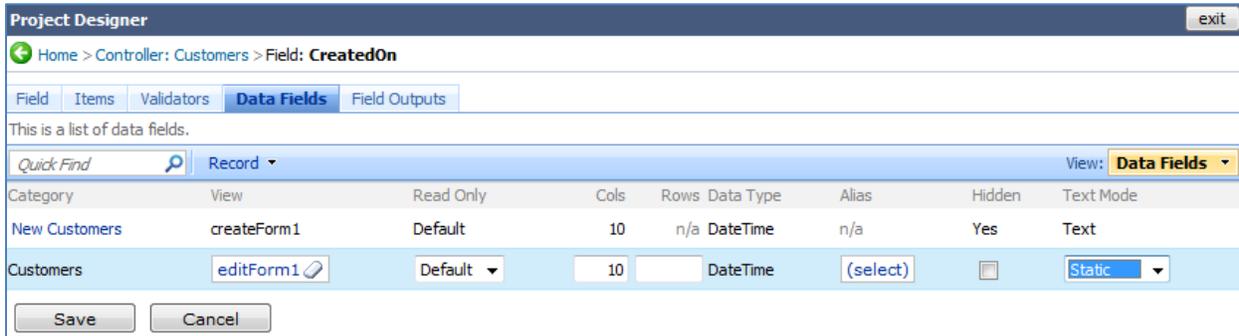
Go back to the list of fields. The *Modified* fields will require an additional modification. Select the *ModifiedBy* field, and switch to *Data Fields* tab. On the action bar, press *New | New Data Field*. Specify the *View* field as "grid1", change *Text Mode* to "Static", and save.
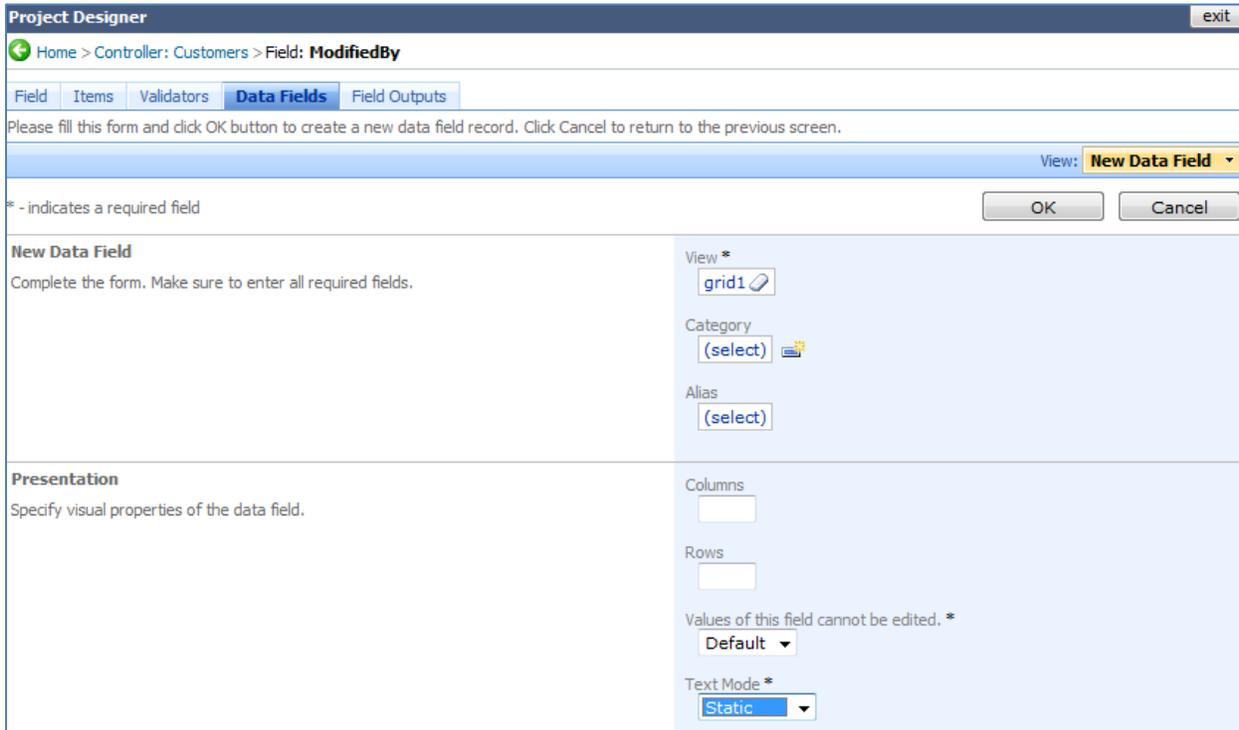
Also, change *createForm1* data field to *Hidden* and change *editForm1* so that *Text Mode* is "Static".

Go back to the list of fields and select *ModifiedOn*. Switch to *Data Fields* tab, and perform the same operations as with *ModifiedBy*.

Exit the *Designer*, and generate the application. When the web page opens, navigate to the *Customers* page. You can see that *ModifiedBy* and *ModifiedOn* fields are now present in grid view. However, if you edit the row in-line, the fields will be read-only.

If you modify a record, you can see that the *Modified* fields will be updated as well.

Now, log out of the application and log in as user (*User Name*: user, *Password*: user123%). Click on *New Customers* button on the action bar. The new fields are not visible in this view.
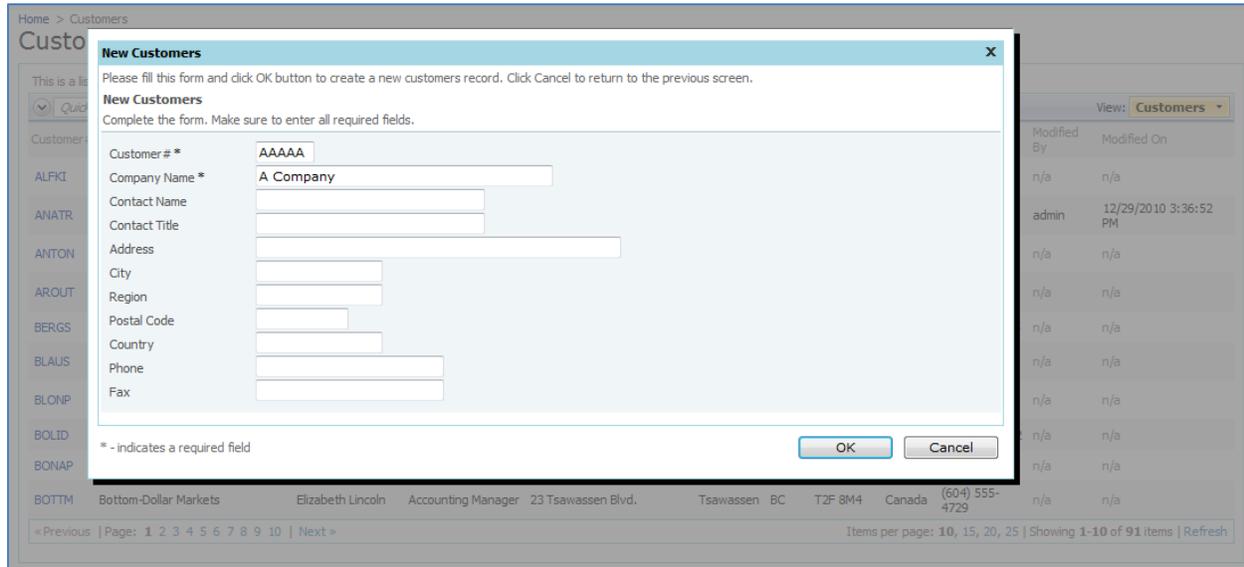
Fill in the required fields, and save the record. It will appear in the *Customers* list, with correct *ModifiedBy* and *ModifiedOn* information. If you select the record, you can view the *CreatedBy* and *CreatedOn* fields in detail view. When you edit the record, the Created and Modified fields will remain in read-only state.
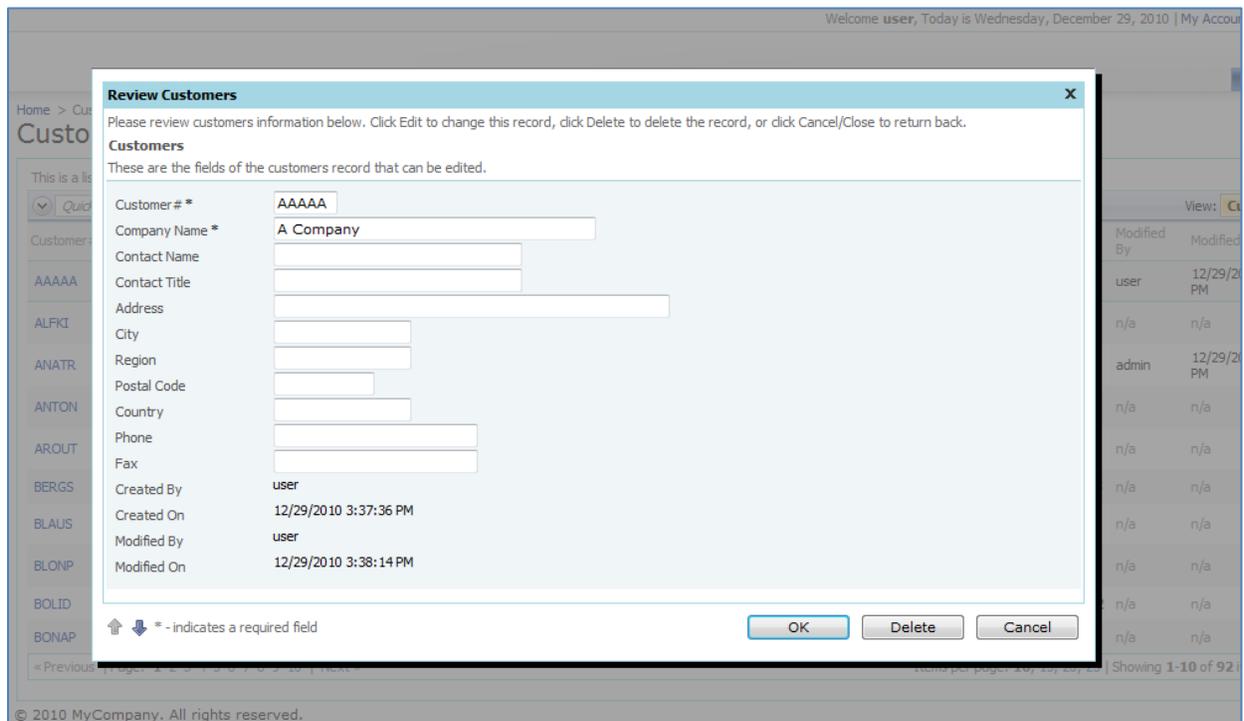
Log out of the application. Log in again as admin. Modify the record you just created, and the *ModifiedBy* and *ModifiedOn* fields will change accordingly. The *CreatedBy* and *CreatedOn* fields will stay the same, however.