

2012



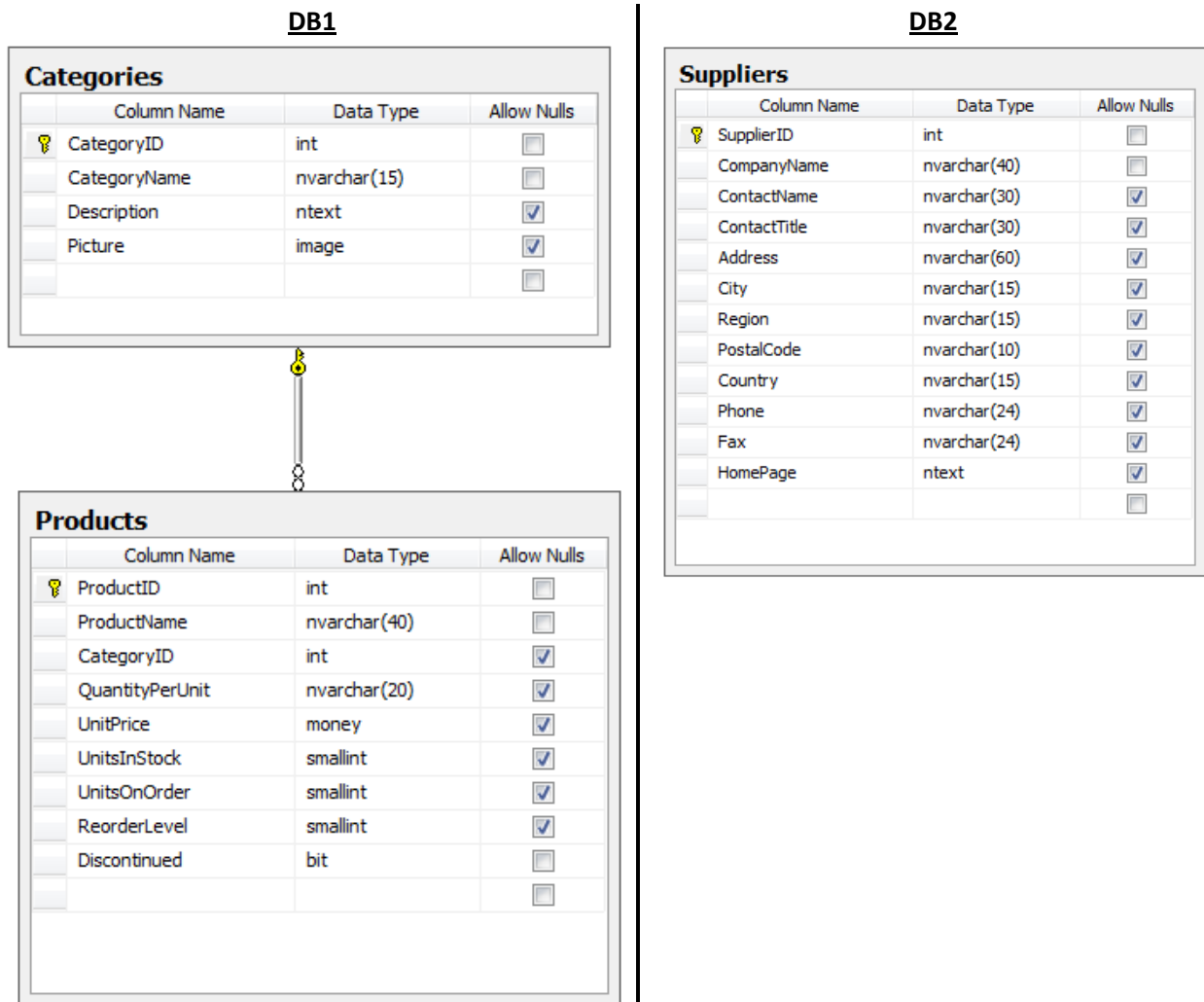
USER GUIDE

Application with Multiple Databases

Application with Multiple Databases

Multiple database services are common in the modern business environment. *Code On Time* generator has the capability to tie this data together in a single web application. This article will entail the steps required to set up such an application.

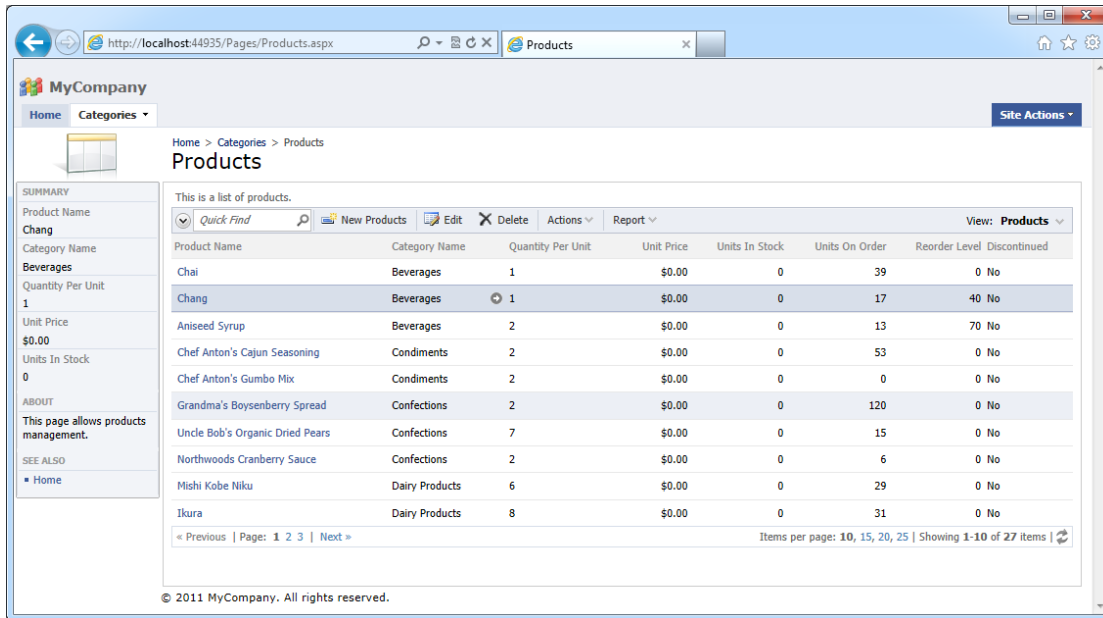
In this example, *DB1* holds the *Categories* and *Products* tables of a web-based order management system. *DB2* holds the *Suppliers* table used by the purchasing department. You can see the database schemas in the pictures below.



Suppose that business requirements demand displaying vendor information next to the product name. We would like to add fields in *Products* to capture information from the *Supplier*, as well as allow access to information about *Suppliers*, all in the same application.

How do we overcome the physical separation of DB1 and DB2?

Let's generate a *Web Site Factory* application from *DB1*. Run *Code On Time Generator*, create a new *Web Site Factory* project by the name of "MultiDB", and press *Create*. Enter the connection string for *DB1*, enable reporting, change the theme to *Social*, and generate the application. You will see that two pages have been created for *Categories* and *Products*, respectively. We will need to add *Suppliers* to this web app as well.



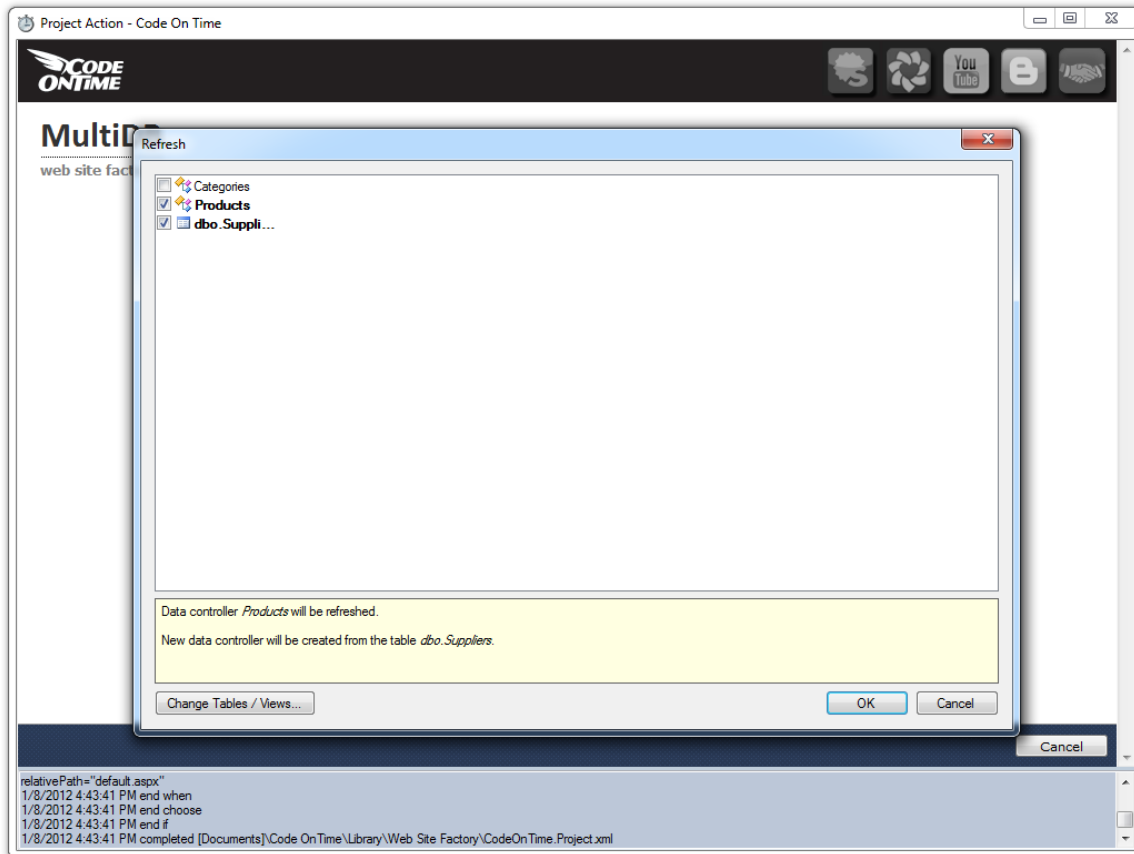
Code On Time Generator does not support generation from multiple databases. Creating an application that handles different data sources will require a combined sample database that you can generate from. After the web app is created, you can change the connection strings to connect the remote data. In this example, it would be easiest to add the *Suppliers* table to *DB1*. Also, add *SupplierID* and *SupplierCompanyName* to the *Products* table. These fields, while not foreign keys in the database, will capture information from *DB2* when we set them up as lookups.

Column Name	Data Type	Allow Nulls
CategoryID	int	<input type="checkbox"/>
CategoryName	nvarchar(15)	<input type="checkbox"/>
Description	ntext	<input checked="" type="checkbox"/>
Picture	image	<input checked="" type="checkbox"/>

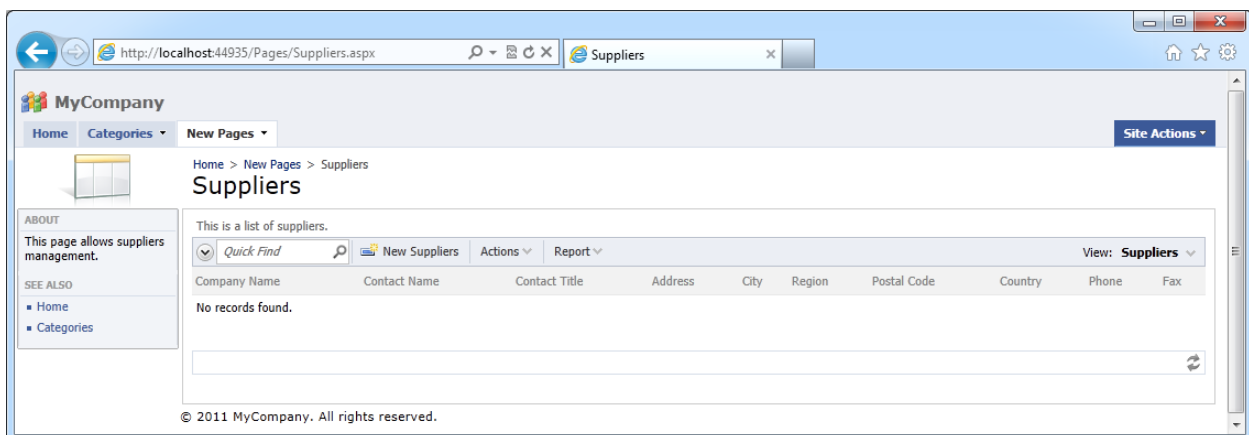
Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
ProductName	nvarchar(40)	<input type="checkbox"/>
SupplierID	int	<input checked="" type="checkbox"/>
SupplierCompanyName	nvarchar(40)	<input checked="" type="checkbox"/>
CategoryID	int	<input checked="" type="checkbox"/>
QuantityPerUnit	nvarchar(20)	<input checked="" type="checkbox"/>
UnitPrice	money	<input checked="" type="checkbox"/>
UnitsInStock	smallint	<input checked="" type="checkbox"/>
UnitsOnOrder	smallint	<input checked="" type="checkbox"/>
ReorderLevel	smallint	<input checked="" type="checkbox"/>
Discontinued	bit	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
SupplierID	int	<input type="checkbox"/>
CompanyName	nvarchar(40)	<input type="checkbox"/>
ContactName	nvarchar(30)	<input checked="" type="checkbox"/>
ContactTitle	nvarchar(30)	<input checked="" type="checkbox"/>
Address	nvarchar(60)	<input checked="" type="checkbox"/>
City	nvarchar(15)	<input checked="" type="checkbox"/>
Region	nvarchar(15)	<input checked="" type="checkbox"/>
PostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
Country	nvarchar(15)	<input checked="" type="checkbox"/>
Phone	nvarchar(24)	<input checked="" type="checkbox"/>
Fax	nvarchar(24)	<input checked="" type="checkbox"/>
HomePage	ntext	<input checked="" type="checkbox"/>

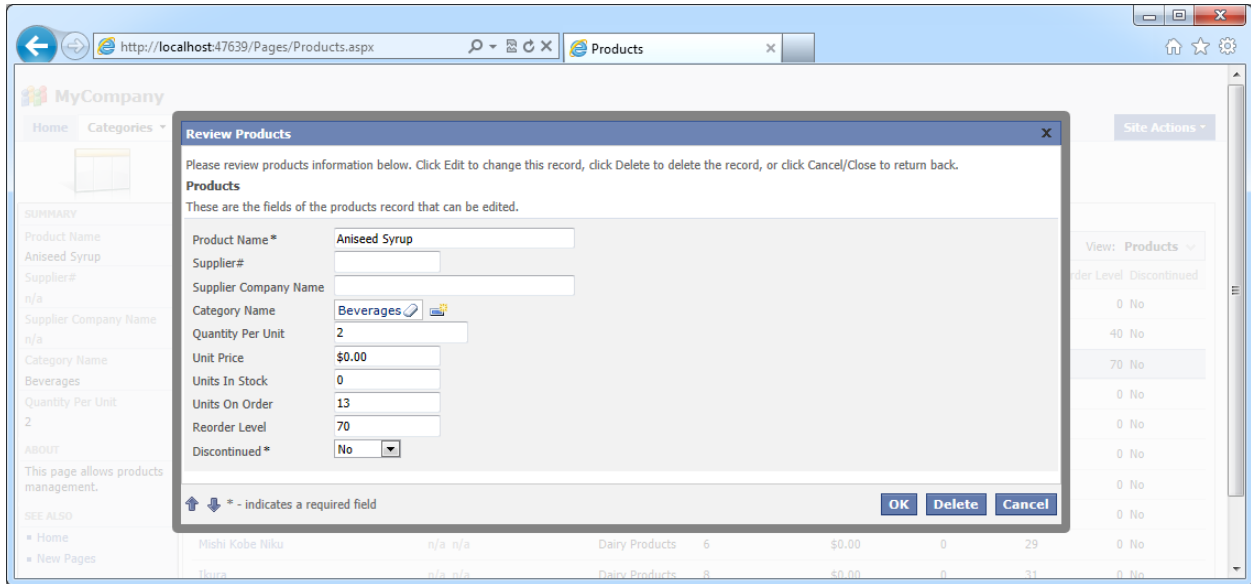
Go back to *Code On Time* and select the project. On the project page, press *Refresh* option. Check *Products* controller and the new *Suppliers* table, and press *OK*. Confirm refresh, and regenerate the application.



When the page loads, you can see that *Suppliers* page has been added under the node *New Pages*. However, there are no suppliers to be found in this database. We will need to change the connection string for the *Suppliers* controller.



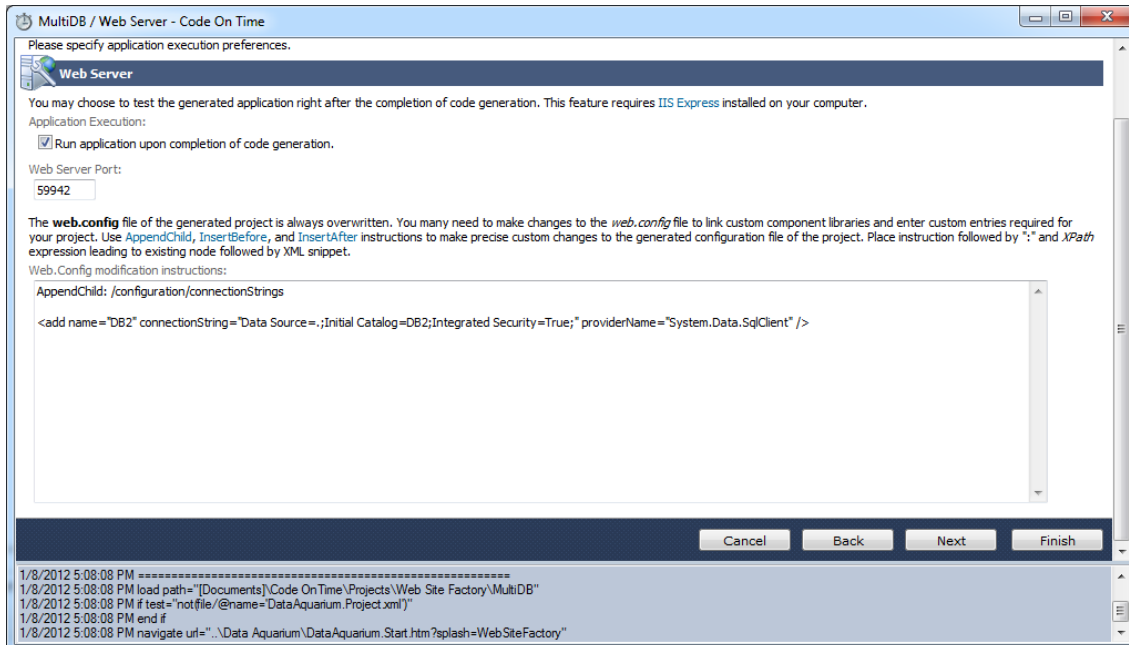
On the *Products* page we will also need to change the *SupplierID* field to a lookup, and use *Supplier Company Name* as its alias.



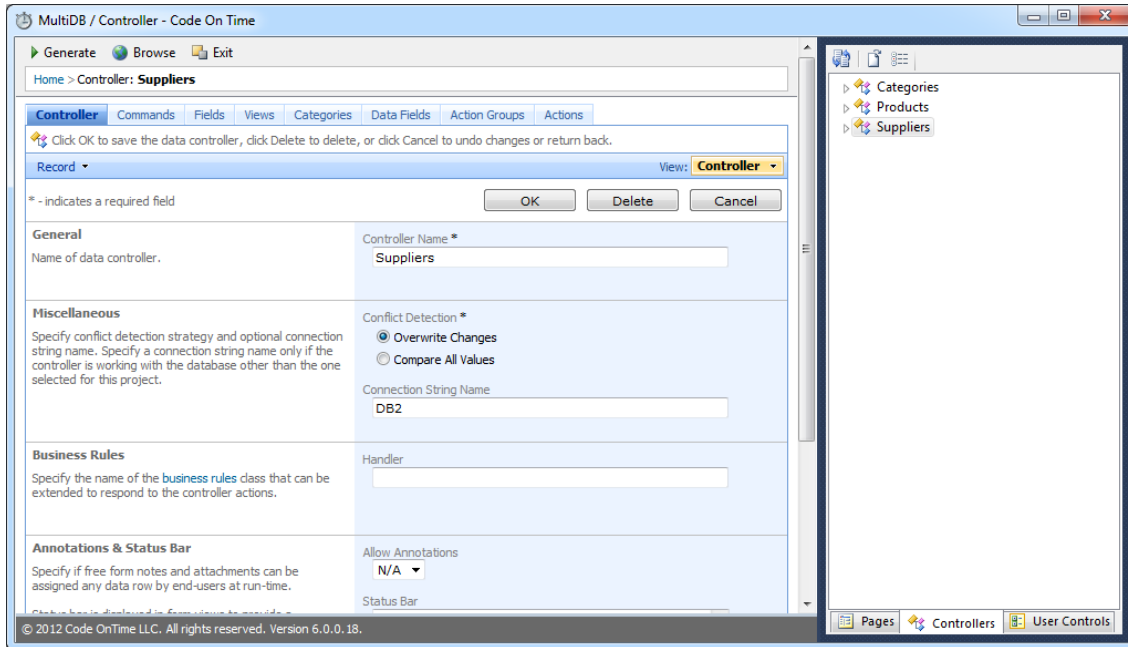
Switch to *Code On Time*, select the project name, press *Settings*, and select *Web Server Configuration*. In the *Web.Config modification instructions* textbox, add the following lines:

```
AppendChild: /configuration/connectionStrings
```

```
<add name="DB2" connectionString="Data Source=.;Initial Catalog=DB2;Integrated Security=True;" providerName="System.Data.SqlClient" />
```

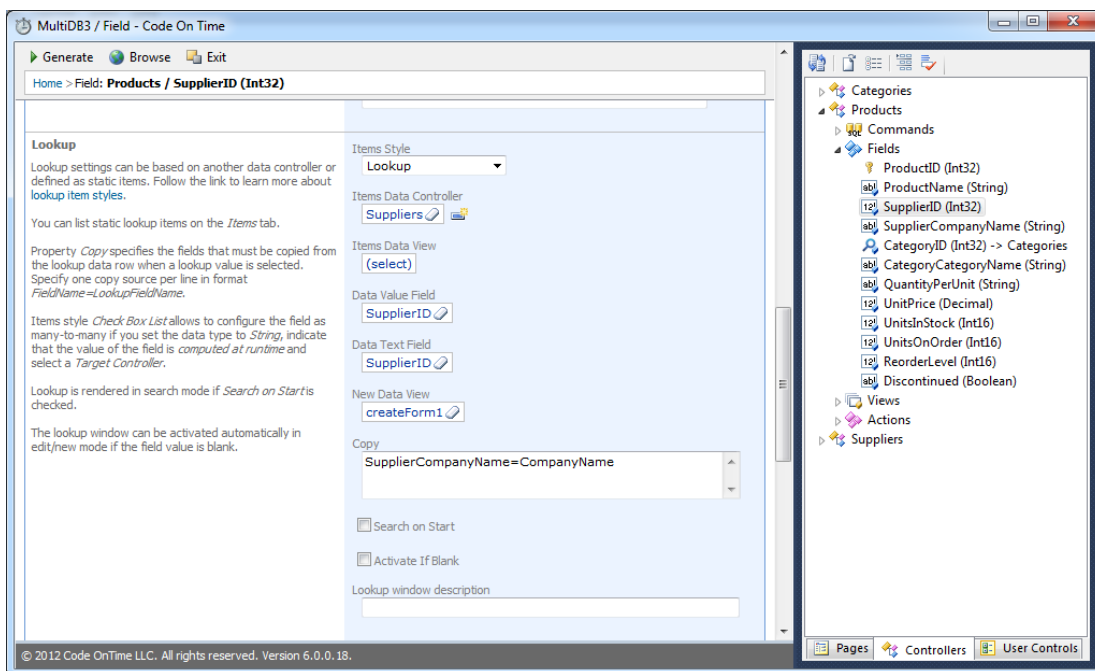


Press *Finish*, and press *Design*. Navigate to the *Suppliers* controller and open its settings. Under *Connection String Name*, type “DB2”.

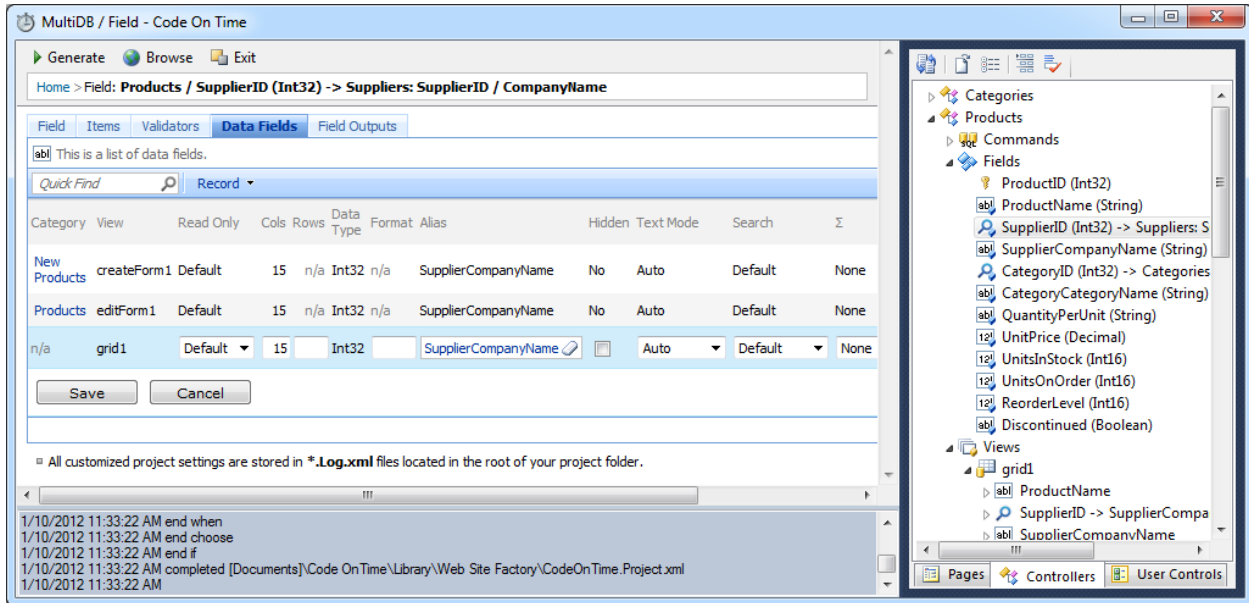


Press *OK* to save your changes. Under the *Products* controller, select *SupplierID* field. In the *Lookup* settings, change *Items Style* to “Lookup”, *Items Data Controller* to “Suppliers”, *Data Value Field* to “SupplierID” and *Data Text Field* to “CompanyName”, and *New Data View* to “createForm1”. In the *Copy* field, type:

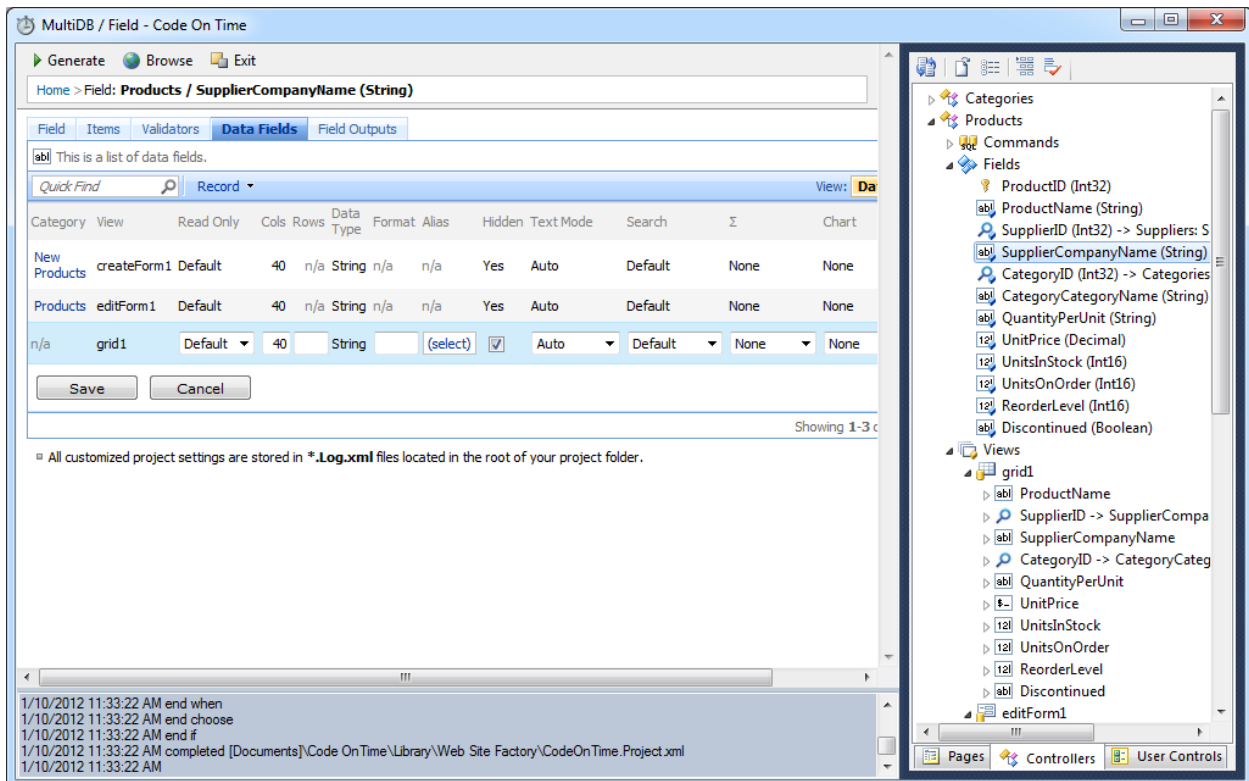
SupplierCompanyName=CompanyName



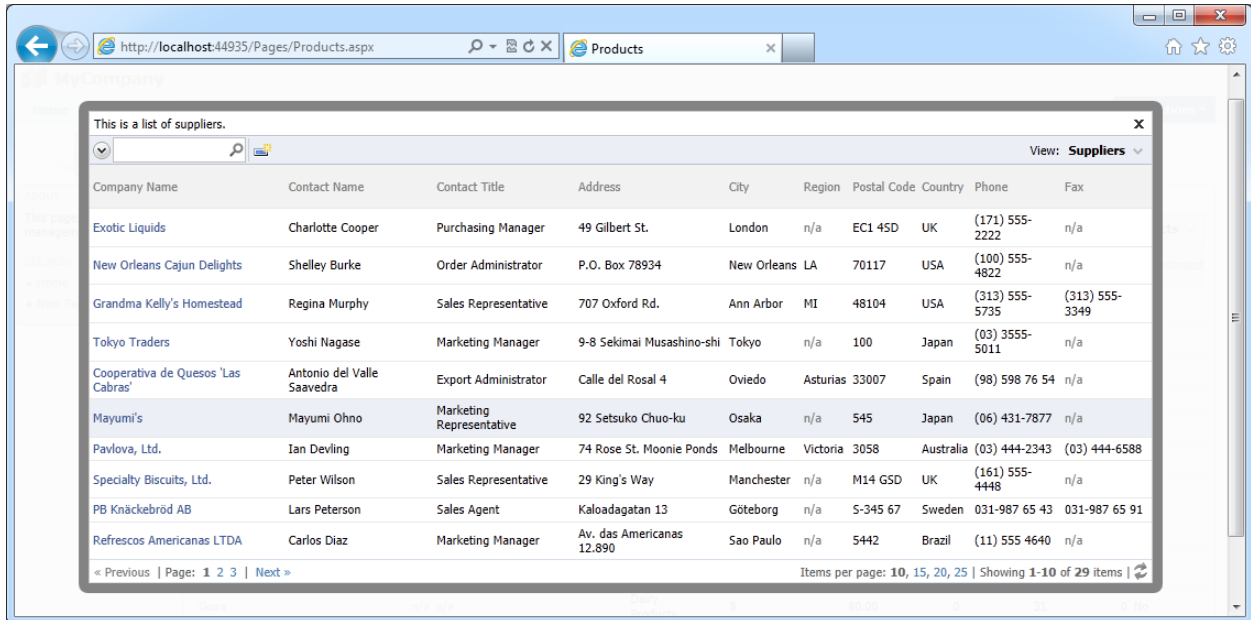
This will ensure that *SupplierID* will lookup data from the *Suppliers* table of DB2, store the *SupplierID* to DB1, and copy the *Company Name* as well. Press *OK* to save. Switch to the data fields tab of *SupplierID* field, and change all the data fields to have an *Alias* of “*SupplierCompanyName*”.



One more change needs to be done. Select the *SupplierCompanyName* field, and switch to the *Data Fields* tab. Change all the data fields to be marked as “*Hidden*”.



Press *Save* to save changes, and generate the project. Go to the *Products* page, and create a new product. You can see that *Supplier ID* is now a lookup. When you click the lookup link, it will show you a list of *Suppliers* from *DB2*.



Select a *Supplier* and you will see that the *Supplier Company Name* will be displayed.

