

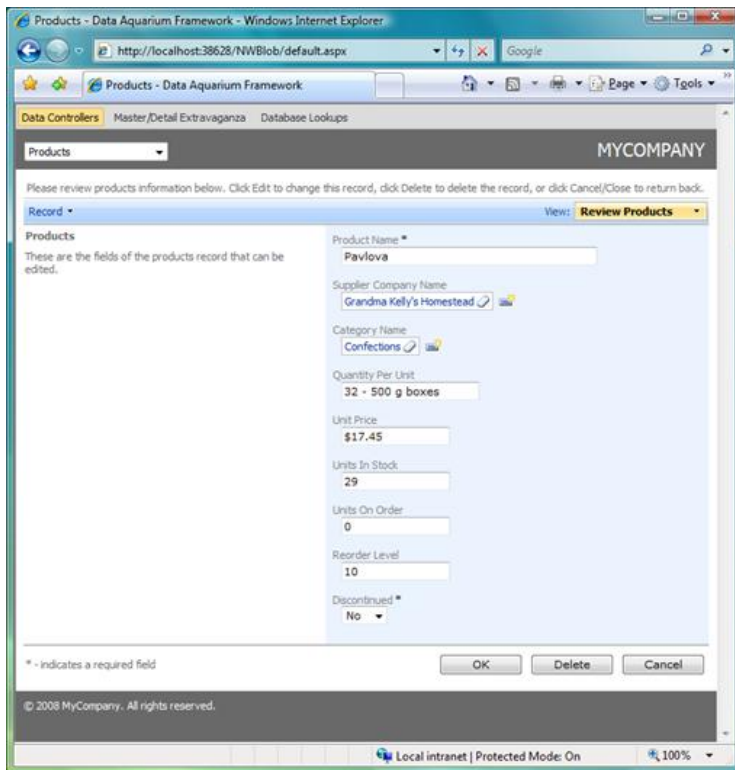
Context-Sensitive Data Lookups

[Data Aquarium Framework](#) now supports context-sensitive lookups.

A common situation in a data entry application may require limiting the possible foreign key values of fields based on other fields in a data table row.

Consider *Northwind* database table *Products*. It stores a product inventory. Each product has a category and a supplier. If you were to build a data entry screen for this table then you may soon realize that it would be nice to limit the number of categories displayed to a user based on a supplier selection. Suppliers typically offer a few categories of products and it may improve data entry efficiency if we were to automatically hide categories that do not have any products that were previously purchased from a supplier.

Here is a picture of a product entry screen automatically created by [Code OnTime Generator](#).



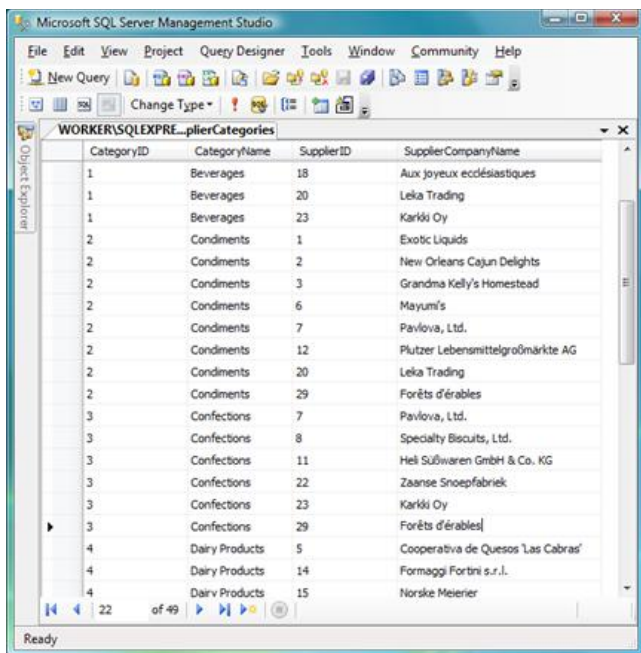
The screenshot displays a web browser window titled "Products - Data Aquarium Framework - Windows Internet Explorer". The address bar shows the URL "http://localhost:38628/NWBlob/default.aspx". The browser window contains a web application interface for "MYCOMPANY". At the top, there is a navigation bar with "Data Controllers", "Master/Detail Extravaganza", and "Database Lookups". Below this is a dropdown menu for "Products" and a "View: Review Products" button. The main content area is titled "Products" and contains a message: "Please review products information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back." Below the message is a form for editing a product record. The form fields are: Product Name (Pavlova), Supplier Company Name (Grandma Kelly's Homestead), Category Name (Confections), Quantity Per Unit (32 - 500 g boxes), Unit Price (\$17.45), Units In Stock (29), Units On Order (0), Reorder Level (10), and Discontinued (No). At the bottom of the form, there are "OK", "Delete", and "Cancel" buttons. A footer note states "© 2008 MyCompany. All rights reserved." and the browser status bar shows "Local intranet | Protected Mode: On" and "100%".

Let's implement a context-sensitive category lookup.

We have to create a database view to allow us implementation of category lookup discussed above. Please execute the following script to create view *SupplierCategories* in the *Northwind* database.

```
create view [dbo].[SupplierCategories]
as
select distinct
    "Categories"."CategoryID" "CategoryID"
    ,"Categories"."CategoryName" "CategoryName"
    ,"Suppliers"."SupplierID" "SupplierID"
    ,"Suppliers"."CompanyName" "SupplierCompanyName"
from
"dbo"."Categories" "Categories"
    inner join "dbo"."Products" "Products" on
        "Products"."CategoryID" = "Categories"."CategoryID"
    inner join "dbo"."Suppliers" "Suppliers" on
        "Suppliers"."SupplierID" = "Products"."SupplierID"
```

The view provide us with all product categories provided by our suppliers.



CategoryID	CategoryName	SupplierID	SupplierCompanyName
1	Beverages	18	Aux joyeux ecclésiastiques
1	Beverages	20	Leka Trading
1	Beverages	23	Karlöö Oy
2	Condiments	1	Exotic Liquids
2	Condiments	2	New Orleans Cajun Delights
2	Condiments	3	Grandma Kelly's Homestead
2	Condiments	6	Mayumi's
2	Condiments	7	Pavlova, Ltd.
2	Condiments	12	Plutzer Lebensmittelgroßmärkte AG
2	Condiments	20	Leka Trading
2	Condiments	29	Forêts d'érables
3	Confections	7	Pavlova, Ltd.
3	Confections	8	Specialty Biscuits, Ltd.
3	Confections	11	Helé Süßwaren GmbH & Co. KG
3	Confections	22	Zaanse Snoepfabriek
3	Confections	23	Karlöö Oy
3	Confections	29	Forêts d'érables
4	Dairy Products	5	Cooperativa de Quesos 'Las Cabras'
4	Dairy Products	14	Formaggi Fortini s.r.l.
4	Dairy Products	15	Norske Meierier

Next we will create a new data controller descriptor *SuppliersCategories.xml* as a copy of *Categories.xml*.

Change command *command1* of new data controller descriptor as shown in this snippet.

```
<command id="command1" type="Text">
  <text>
    <![CDATA[
select
  "Categories"."CategoryID" "CategoryID"
  ,"Categories"."CategoryName" "CategoryName"
  ,"Categories"."Description" "Description"
  ,"SupplierCategories"."SupplierID" "SupplierID"
  ,"SupplierCategories"."SupplierCompanyName" "SupplierCompanyName"
from "dbo"."Categories" "Categories"
  inner join "dbo"."SupplierCategories" "SupplierCategories" on
  "SupplierCategories"."CategoryID" = "Categories"."CategoryID"
]]>
  </text>
</command>
```

Change *fields* element to provide definitions for fields *SupplierID* and *SupplierCompanyName*.

```
<fields>
  <field name="CategoryID" type="Int32" allowNulls="false"
isPrimaryKey="true" label="Category#" readOnly="true" />
  <field name="CategoryName" type="String" allowNulls="false" label="Category
Name" />
  <field name="Description" type="String" allowQBE="false"
allowSorting="false" label="Description" />
  <field name="SupplierID" type="Int32" label="Supplier#">
  <field name="SupplierCompanyName" type="String" label="Supplier Company
Name"/>
```

```
</fields>
```

Change view *grid1* to include field *SupplierID* under alias *SupplierCompanyName*.

```
<dataFields>
  <dataField fieldName="CategoryName" columns="15" />
  <dataField fieldName="Description" rows="5" />
  <dataField fieldName="SupplierID" aliasFieldName="SupplierCompanyName"/>
</dataFields>
```

The purpose of these changes is to allow filtering by *SupplierID* when grid view *grid1* is presented to a user.

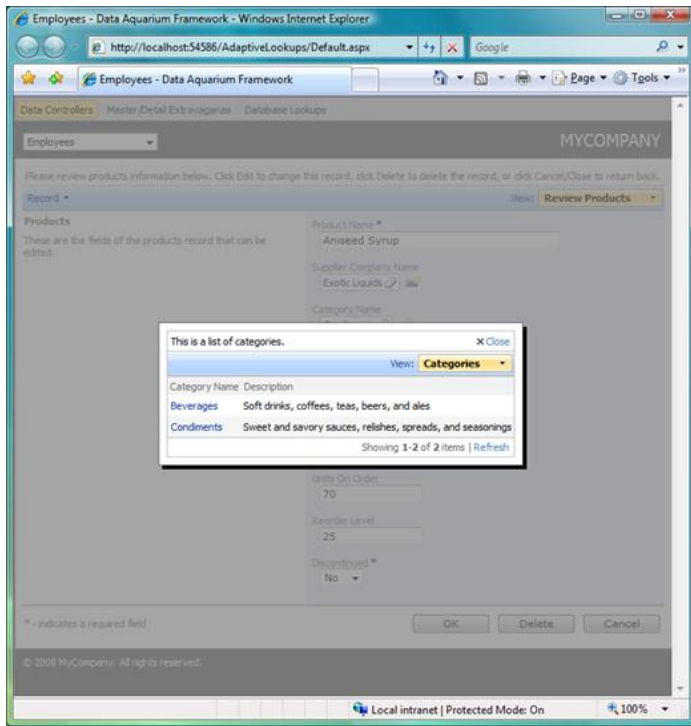
Now we are ready to modify the data controller descriptor *Products.xml* to use *SupplierCategories* instead of *Categories* in the lookup definition of field *CategoryID*.

```
<field name="CategoryID" type="Int32" label="Category#"
sourceFields="SupplierID">
  <items style="Lookup" dataController="SupplierCategories"
newDataView="createForm1" />
</field>
```

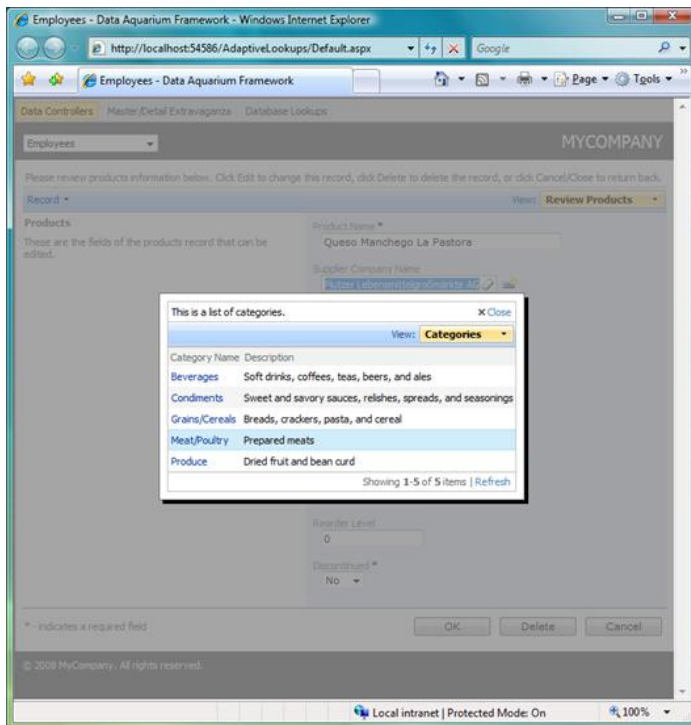
We have replaced data controller reference from *Categories* to *SupplierCategories*.

Another important change is a new attribute *sourceFields*. This attribute specifies a comma-separated list of fields defined in this data controller descriptor that will provide the source of filter values for the fields in the view of *SupplierCategories* data controller that have matching names when a lookup is displayed. The filter is automatically set as an external, which hides the filter field from end-user. *SupplierID* is never presented by is used in filtering.

Here is how it looks in a real application. Only two categories of products are available from supplier *Exotic Liquids*.



Let's select supplier *Plutzer Lebensmittelgroßmärkte AG* and check out the available categories. Five categories are available for selection.



Context-sensitive lookups are available to [premium project subscribers](#).