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 where 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.
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.

```sql
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.
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.

```xml
<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`.

```xml
<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"/>
```
Change view grid1 to include field SupplierID under alias SupplierCompanyName.

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.

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.