Example of using NSBrowser with a Core Data Application

R.R. Bradley
US Meat Animal Research Center, Clay Center, Nebraska 68933, USA

Here is an example of how one might use NSBrowser and NSTreeController with a Core Data document based application. For this example, we will use a country, state, county hierarchy.

1. In Xcode create a new Cocoa Application selecting:
    a. "Create document-based application"
    b. "Use Core Data for storage"
2. In the "Models" folder, edit "MyDocument.xcdatamodel"
3. Create a new entity named "Country"
    a. Add a non-optional attribute "name". †NSTreeController requires "Key Value Coding" compliant entities. Simply put, this means you must use the same attribute name for all entities in the hierarchy.
    b. Set:
        i. type to string
        ii. default value to "Unnamed"
    c. Add an optional string attribute "capital".
4. Create a new entity named "State"
    a. Add a non-optional attribute "name".
    b. Set:
        i. type to string
        ii. default value to "Unnamed"
    c. Add an optional string attribute "capital".
    d. Add a relationship from the State entity to the Country entity.
        i. Set name to "parent"
        ii. Unset "optional"
5. Add a relationship from the Country entity to the State entity.
        i. Set name to "children".
        ii. Set Optional checkbox.
        iii. Set To-many checkbox.
        iv. Set "Inverse:" to parent
6. Create a new entity named "County"
    a. Add a non-optional attribute "name".
    b. Set:
        i. type to string
        ii. default value to "Unnamed"
    c. Add an optional string attribute "county_seat".
    d. Add a relationship from the County entity to the State entity.
        i. Set name to "parent"
        ii. Unset "optional"
7. Add a relationship from the State entity to the County entity.
        i. Set name to "children".
        ii. Set Optional checkbox.
        iii. Set To-many checkbox.
        iv. Set "Inverse:" to parent

That's it for the data model. It should look something like figure 1.

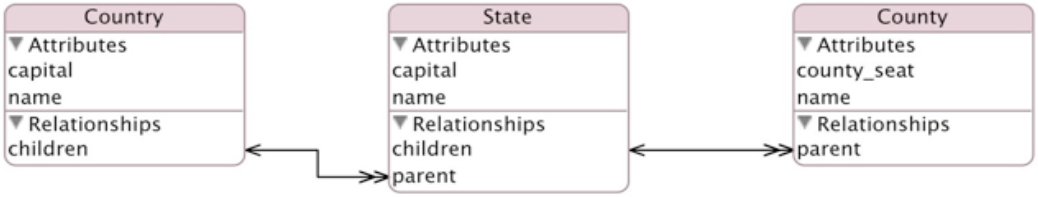

Figure 1

Next we need to build the user interface. Open the "Resources" folder and double click "MyDocument.xib" to launch Interface Builder and start editing the xib file.

Change the text from "Your document contents here" to "Country Browser" and drag it to the top of the window. Set the "anchor" to anchor it to the top center of the window. Drag an NSTreeController object from the library over to the MyDocument.xib window. From the inspector, set attributes:

1. "Tree Controller":
    a. Children = children
    b. Leaf = leaf
2. "Object Controller"
    a. Mode = Entity
    b. Entity Name = Country
    c. Prepares Content = checked

Then set binding parameters:

- Managed Object Context
    a. Bind to: "File's Owner"
    b. Controller Key leave blank
    c. Model Key Path = managedObjectContext

Drag over an NSBrowser object to the window. Navigate to the "bindings" tab and down to the "Browser Content" section.

1. Bind "Content" to "Tree Controller"
    a. Set "Controller Key" to "arrangedObjects"
    b. Leave "Model Key Path" and "Value Transformer" blank
2. Bind "Content Values" to "Tree Controller"
    a. Set "Controller Key" to "arrangedObjects"
    b. Set "Model Key Path" to "name"
    c. Leave "Value Transformer" blank
3. Bind "Selection Index Paths" to "Tree Controller"
    a. Set "Controller Key" to "selectionIndexPaths"
    b. Leave "Model Key Path" and "Value Transformer" blank

Next drag over 3 buttons of your choice. I used "Bevel Buttons" and set the "Image" to "NSAddTemplate" for two of the buttons and gave them labels of "Add" and "Add Child". For the third button I set the "Image" to NSRemoveTemplate" and gave it a label of "Remove".

1. Control drag from the "Add" button to "Tree Controller"
    a. wire the connection to "Add:"
    b. From the bindings tab "Availability" section, bind "Enabled" to "Tree Controller"
        i. Set "Controller Key" to "canAdd"
        ii. Leave "Model Key Path" and "Value Transformer" blank
2. Control drag from the "Add Child" button to "Tree Controller"
    a. wire the connection to "addChild:"
    b. From the bindings tab "Availability" section, bind "Enabled" to "Tree Controller"
        i. Set "Controller Key" to "canAddChild"
        ii. Leave "Model Key Path" and "Value Transformer" blank
    c. From the bindings tab "Availability" section, bind "Enabled2" to "Tree Controller"
        i. Set "Controller Key" to "selection"
        ii. Set "Model Key Path" to "leaf"
        iii. Set "Value Transformer" to NSNegateBoolean
3. Control drag from the "Remove" button to "Tree Controller"
    a. wire the connection to "remove:"
    b. From the bindings tab "Availability" section, bind "Enabled" to "Tree Controller"
        i. Set "Controller Key" to "canRemove"
        ii. Leave "Model Key Path" and "Value Transformer" blank

That's about it for this exercise. We can now build and run the application. The document window of the running application should look something like figure 2 on the following page.

Figure 2

Obviously we cannot do much at this stage of development. However, it is surprising how much we can do without any coding. In order to add some utility to this application, we would need to add a delegate to provide some useful data.