

MasteringDetails

by Craig Federighi, EO Development Team

Overview

This example presents a generic scheme for assigning relationship keys automatically for insertions and deletions in detail controllers. The implementation of this is fairly complex, but the result is a significant simplification in the code required to implement applications that allow editing master-detail relationships.

Demo Application Features

The application allows the user to enter and edit employee information. The employee data has several interesting relationships:

- A to-one foreign key to primary key relationship for employee to department.
- A many-to-many relationship between employee and project.
- Primary key to primary key relationship for Quote and Picture.

The primary purpose of this example is to show how key assignment can be handled in a general way when records are inserted in detail controllers with these sort of relationships.

Additional Features:

- The Quote field demonstrates storing rich text in the database.
- An image well allows storage and retrieval of images.
- Departments and Projects are selected by selecting from a list of available choices

- presented in a scrolling list (pick list). This is done with a new association type.
- New objects are automatically assigned unique primary keys.

The Example Folder

The example contains the following:

- EOFExtensions.subproj - A application-independent relationship management facility.
 - KeyGenerator - Much like the UniqueKey example, this provides a facility for assigning unique primary keys to objects.
 - ValueForKey - A category on NSObject to provide single key versions of takeValuesFromDictionary and valueForKey.
 - RelationshipKeySetter - The brains of this example. Provides categories on EORelationship to perform key assignment on master or detail objects in order for the relationship between them to hold in the database.
 - SavvyControllerDel... - A generic EOController delegate that uses the RelationshipKeySetter to assign keys correctly when objects are inserted or deleted in detail controllers.
 - SelectionInsertionAs... - An EOAssociation subclass that inserts and deletes objects in its controller based on the selection in its destination controller.
 - ResetRelationships - Not actually used in this example. Provides a category on EODatabaseChannel for updating the relationship pointers of an EO when its foreign key values have been modified. This plays the opposite role to the RelationshipKeySetter.
- EmpController - The main controller in this application. Sets up associations, performs initial fetches, and provides an insert operation to add new employees. As a subclass of the generic SavvyControllerDelegate

it acts as a delegate for detail controllers, performing key assignments for the insertion and deletion of objects.

Employee - An example EO class.

Running the Example

Select an employee from the table view; notice that the department and project lists at the bottom of the window select the department and projects for the selected employee. You can edit the employee in several ways:

- Re-assign an employee to a different department by clicking on the new department in the departments list.
- Assign the employee to additional projects by shift clicking on a new project in the projects list.
- Add a picture for the employee by dragging an image into the image well.
- Type a new "quote" for the employee by typing in the Quote text view. This is an RTFD field, so you can use rich text and even add images or files.

Save your changes by either clicking the save button or selecting a different employee.

Add a new employee by clicking the Add button. Enter the new data for this employee and select save to commit the new entry to the database.

Building For Sybase v.s. Oracle

This example uses the "People" database. Scripts to create this database on your server can be found in **/NextDeveloper/Examples/EnterpriseObjects/PeopleDBScripts**. This example includes model files for both Sybase and Oracle. To use Oracle, do the following in a shell:

cp people_Oracle.eomodel people_photo_quote.eomodel
and then make the project.

Known Limitations

The SavvyDelegate performs non-optimal database updates, sometimes updating an object twice or inserting an object and immediately updating it as part of a single save operation.

Possible Enhancements

The SavvyDelegate does not handle propagating deletions from a master to its details. It would be nice if we could just delete an employee from the master and have other associated data dropped automatically.

This app performs no validation on employee data. Combining the SavvyDelegate here and the ValidatingDelegate from the Validation example could improve that situation.

Philosophical Reflections

This example shows how to eliminate a lot of the drudgery associated with assigning keys in detail controllers. One might ask why the assignment is being performed at the controller level at all. For example, what if I create an employee object on a screen that does not display the department or employee picture (and therefore doesn't have detail controllers for these properties)? I'd still like the employee to automatically be assigned a default department and (possibly empty) picture and quote objects.

A better (and less UI-centric) way to handle the task of relationship management is to perform the work of the Savvy(Controller)Delegate in a delegate of the EODatabaseChannel, and/or in the EO itself. This is left as an exercise for the reader!

Change History

17/Feb/95 Created for EOF 1.1 release (Craig Federighi)