Learning EOModeler

by Theresa Ray of Tensor Information Systems, Inc.

Sponsored by Apple Computer, Inc.
Apple Developer Connection

Learning EOModeler

by Theresa Ray of Tensor Information Systems, Inc.

Learning how to effectively use the EOModeler application is an essential beginning to the development of any database driven web site. This survival guide explains the most commonly used features and options available in the EOModeler application.

One of the first steps you take in development of a database driven web site is to create an EOModel. A model defines the mapping between your enterprise objects and the data stored in your database. An enterprise object class typically corresponds to a table in a database, and an enterprise object instance corresponds to a single row or record in the corresponding table. You define and store this correspondence in models that you build graphically (an EOModel) with the EOModeler application.

This survival guide steps through an example EOModel, defining the menu options and symbols you see when using the EOModeler application. After WebObjects is installed, an example named Movies can be located in the < APPLE_ROOT> \Developer\Examples\WebObjects\Java directory. The EOModel used in the Movies example application (named Movies.eomodeld) forms the basis of the example used in this guide. You may find it helpful to have the Movies EOModel open on your computer while you read this guide. It would also help if you understand something about Entity-Relationship modeling when reviewing this guide. If you aren't familiar with E-R modeling, you may want to read the Appendix, "Entity-Relationship Modeling" in the Enterprise Objects Framework Developer's Guide first.

Model Information & Database Connection
The left-hand column contains the name of the EOModel at the top (Movies). Underneath the EOModel name is a list of all entities in this model (Director, Movie, MovieRole, etc). If an entity has one or more relationships defined to another entity, the '+' symbol will appear next to the entity name in the column on the left. Clicking on the '+' expands the entity-relationship by one level.

Clicking on the EOModel name and clicking on the toolbar icon symbolized with a circle-I (or choosing the "Tools" menu option and selecting "Inspector…") opens the Connection Dictionary Inspector. This inspector allows you to review the username, password, database name and other options used by the Enterprise Objects Framework when connecting to the database. To change these options, choose the "Model" menu option and select "Set Adaptor Info…". If you need to change the type of database you are connecting to altogether (i.e. from Informix to Oracle), choose the "Model" menu option and select "Switch Adaptor…".

Going back to the Inspector panel and clicking on the right tab (the open book symbol) opens the EOModel UserInfo Inspector, where you can review any user-defined information about your model (by default, the modification history is stored here).

Entity Definition
In the main EOModel window, the right-hand section displays columns containing details about each entity. The first column displays the entity name, which is user-defined and independent of both the database table name and the class name.

The second column displays the name of the corresponding table in the database (some databases like Informix require the table name to be a case-sensitive match with the database).

The third column displays the class name for that entity. When you add a new entity, its class is set by default to EOGenericRecord. If you do not need to add business logic or specific behavior to your class, you may leave the class name as EOGenericRecord. If you create a custom class for your entity, the name for your class will be represented here (for example, Movie).

Clicking on the row detailing the Movie class in the right-hand column, then clicking on the circle-I (or choosing the "Tools" menu option and selecting "Inspector…") opens the Entity Inspector. This panel lists the properties for the selected entity and displays the attribute's current settings (more information on those settings is provided in the following sections).

Clicking on the second tab at the top of the Inspector panel (symbolized by a half-full disc) updates the Inspector to display the Advanced Entity Inspector. This is where you define the batch faulting size for your entity, where you can define an external query, set a qualifier, set the parent class, make the entity read only, set the entity to cache in memory, or define the entity as an abstract class. The Enterprise Objects Framework Developer's Guide provides detailed documentation on each of these options.

Clicking on the third tab at the top of the Inspector panel (symbolized by a database cylinder) opens the Stored Procedure Inspector. This panel allows you to review and update the settings corresponding to any stored procedures you may have defined in the database. In this example, there are no defined stored procedures.

Clicking on the fourth tab at the top of the Inspector panel (symbolized by an open book) opens the EOEntity UserInfo Inspector, where you can review any user-defined information about your entity.

Returning to the main EOModel window, you can add columns to the default display in the right-hand view by clicking on the "Add column…" pulldown at the bottom of this section. You can reorder column display by clicking on the column title and dragging the column to the appropriate place.

Attributes

Double-clicking on the icon next to the 'Movie' entity in the right-hand column (or the Movie disc in the left-hand column) changes the right-hand view to show the attributes and relationships associated with the Movie entity. The top half of the right-hand section displays the attributes for this entity (category, dataReleased, language, etc). Each attribute has several columns associated with it that define the settings for the attribute.

The column with the key defines the primary key for this entity. The key may be compound (made up of multiple columns). Click in the box under the key column next to the desired attribute(s) to define the entity's primary key.

The column with the diamond identifies attributes that are meant to be class properties – accessible by applications using this model. If there is a diamond displayed in the column next to an attribute, your application will be able to access the value stored in the corresponding column in the database. Clicking in the box next to the attribute in this column toggles the class property nature of the attribute on and off. Foreign keys are usually not class properties, and in many cases the primary key is also not needed as a class property. Columns unused by your application should not contain a diamond in order to keep the snapshot small.

The next column shows a padlock. Attributes displaying a padlock are used in whatever locking strategy you choose for your model (locking strategy options are detailed in the Enterprise Objects Framework Developer's Guide). In the default case of optimistic locking, the Enterprise Objects Framework compares the value of the snapshot stored when an object is initially fetched with its current database value for attributes marked with a padlock. If the value for an attribute marked with a padlock changes, EOF will react however the programmer has determined (the default is to raise an exception). Locking strategy choice affects exactly how EOF will respond when a locking conflict is determined.

The next column defines the attribute name. The attribute name is the name by which your application will access the data for the corresponding row in the database, but the attribute name does not need to correspond to the attribute name as defined in the database.

The next column defines the value class for the attribute. The value class defines what object type EOF will convert the value to/from when fetching or saving.

The next column defines the external type for the attribute – how the attribute is defined in the database. This is necessary so that EOF knows how to convert the value to/from the appropriate value class.

The next column defines the database-defined width of the attribute if appropriate. For example, if you have defined the attributed named category in your database as a varchar(60), the external type should be set to char or varchar, and the width to 60.

You can add columns to the default display in the right-hand view by clicking on the "Add column…" pulldown at the bottom of this section. You can reorder column display by clicking on the column title and dragging the column to the appropriate place.

Clicking on an attribute and clicking on the icon symbolized with a circle-I (or choosing the "Tools" menu option and selecting "Inspector…") opens the Attribute Inspector. The Attribute Inspector lets you update all the settings for an attribute, whether or not the detail is displayed in the entity's attribute view.

The Attribute Inspector's main panel lets you define the attribute name, the external database column name (usually case insensitive) or derived information, and the external type. You can also specify the Internal Data Type (this is the value class), and the external width, precision, or scale if appropriate.

Clicking on the second tab in the inspector (symbolized by an eighth of a disc) opens the Advanced Attribute Inspector. Here you can update whether the attribute is read-only, whether it allows null values, and define any custom formatting.

Clicking on the third tab brings in the inspector (symbolized by an open book) opens the Attribute UserInfo Inspector, where you can review any user-defined information about your attribute.

The entire row for the attribute will be in bold if the attribute is a derived value or a flattened attribute.

Relationships
In the main EOModel window on the right side and underneath the attribute section is the relationship section. This section displays the details for relationships for this entity.

The first column displays '>' if the relationship has been defined as a one-to-one relationship, and '>>' if the relationship has been defined as a one-to-many relationship. Clicking on the '>' or '>>' traverses the relationship - that is the view on the right-hand side of the model's display will change to reflect the attributes and relationships of the destination entity for that relationship.

The second column, symbolized by a diamond, indicates whether the relationship is a class property (in a many-to-many relationship, you will have an intermediate association table and you may not require the relationship from this entity to the association table to be accessible by your class). Clicking in the box next to the attribute in this column toggles the class property nature of the relationship on and off.

The next column is the relationship name. The relationship name is the name by which your application will make the foreign-key join with the destination entity in the database. For example, if you wanted to retrieve all the rows from the director table that are foreign-key related to a specific row for the movie, your application would access the instance variable "directors".

The next column is the destination entity name. This is the name of the entity for which the current entity has a foreign-key or primary-key relationship.

The final two columns detail the source and destination attributes that define how the two entities are related.

You can add columns to the default display in the right-hand view by clicking on the "Add column…" pulldown at the bottom of this section. You can reorder column display by clicking on the column title and dragging the column to the appropriate place.

Clicking on a relationship and clicking on the icon symbolized with a circle-I (or choosing the "Tools" menu option and selecting "Inspector…") opens the Relationship Inspector. The Relationship Inspector lets you update all the settings for a relationship, whether or not the detail is displayed in the entity's relationship view.

The Relationship Inspector's main panel lets you define the relationship name, the model in which the destination attribute resides (you may define a relationship to an entity from another model), the destination entity name, the relationship type (to-one or to-many), and the source and destination attributes that should be used to define the join (you can connect multiple sets).

Clicking on the second tab in the inspector (symbolized by two half discs) opens the Advanced Relationship Inspector where you can define the batch faulting size, whether the destination is optional or mandatory, the delete rule action, and select whether this entity owns the destination entity and/or whether this entity should propagate its primary key to the destination entity (the Enterprise Objects Framework Developer's Guide explains these options in detail).

Clicking on the third tab brings in the inspector (symbolized by an open book) opens the Relationship UserInfo Inspector, where you can review any user-defined information about your relationship.

You may flatten an attribute or relationship from the destination of a relationship to the current entity. Flattening an attribute or relationship into the current entity allows the attribute or relationship to appear to your application as if it were a basic part of the current entity. EOF automatically handles traversing the relationship and returning the appropriate values for you. To flatten a property, double-click on the '>' or '>>' at the left of the relationship

to traverse the relationship from the current entity. Select the attribute or relationship you wish to flatten, then choose the menu option "Property" and select "Flatten Property".

The entire row for the relationship will be in bold if the relationship is a derived value or a flattened relationship.

Quick-keys
There are many quick-keys provided in the EOModeler application. These are displayed as icons across the top of the panel – just under the menu options.

The icon symbolized by a hierarchical diagram lets you change the display option for the model to browser view, diagram mode or table view. Diagram mode shows the model in a traditional database schema diagram view. Table view shows only the entity, attribute and relationship names and their organization within the model. The view may also be changed by choosing the "Tools" menu option and selecting "Browser Mode", "Table Mode" or "Diagram View". Changing the view may change the available quick-keys, depending on the information currently displayed in the main EOModel window.

The icon symbolized by two pages with title bars lets you open a new EOModel editor window. This may also be accomplished by choosing the "Model" menu option and selecting "New".

The icon symbolized by two pages without title bars lets you copy whatever items are selected in the model. This includes attributes, relationships, entities, or just data. This may also be accomplished by choosing the "Edit" menu option and selecting "Copy".

The icon symbolized by a pair of scissors lets you cut whatever items are selected in the model. This includes attributes, relationships, entities, or just data. This may also be accomplished by choosing the "Edit" menu option and selecting "Cut".

The icon symbolized by a clipboard lets you paste any previously cut or copied attributes, relationships, entities, or data. This may also be accomplished by choosing the "Edit" menu option and selecting "Paste".

The icon symbolized by a page with a red X across it lets you delete whatever items are selected in the model. This includes attributes, relationships, entities, or just data. This may also be accomplished by choosing the "Edit" menu option and selecting "Delete".

The icon symbolized by a counterclockwise arrow lets you undo the last edit. This may also be accomplished by choosing the "Edit" menu option and selecting "Undo".

The icon symbolized by a database cylinder and magnifying glass opens the data browser. By default the data browser will fetch the values stored in the database for whatever entity is selected. You can change the sort ordering of the selected items by clicking on the pyramids in the bottom right-hand corner of the window. You can change the order of the columns by clicking the column title and dragging it to the desired location. You can also specify a qualifier for the fetch, or force a refetch of the items from the database. This may also be accomplished by choosing the "Tools" menu option and selecting "Data Browser…".

The icon symbolized by a page and "SQL" opens a window containing the SQL required to administer the table in the database associated with the selected entity. The checkboxes within this window alter the SQL generated by

EOF and enable you to create the exact administrative SQL options you need. This window may also be reached by choosing the "Property" menu option and selecting "Generate SQL…".

The icon symbolized by two pages and "m" generates the Objective-C for a custom class implementation of the selected entity (you will be prompted for class name and destination directory). The Objective-C generated by this option contains the skeleton functionality for the class implementation including instance variable definitions and accessor methods. This may also be accomplished by choosing the "Property" menu option and selecting "Generate ObjC Files…".

The icon symbolized by a page and "jav" generates the Java for a custom class implementation of the selected entity (you will be prompted for class name and destination directory). The Java generated by this option contains the skeleton functionality for the class implementation including instance variable definitions and accessor methods. This may also be accomplished by choosing the "Property" menu option and selecting "Generate Java Files…".

The icon symbolized by a disc and "+ " adds a new entity to the model. This may also be accomplished by choosing the "Property" menu option and selecting "Add Entity".

The icon symbolized by a eighth of a disc and "+ " adds a new attribute to the selected entity. This may also be accomplished by choosing the "Property" menu option and selecting "Add Attribute".

The icon selected by two discs and "+ " adds a new relationship to the selected entity. This may also be accomplished by choosing the "Property" menu option and selecting "Add Property".

The icon symbolized by a disc, a page and "+ " adds a new fetch specification to the selected entity (new in WebObjects 4.0). Clicking on this icon changes the right-hand view of the model to display the details associated with the fetch specification. You may name the fetch specification using the text field at the top of this view. To define the qualifier, select the attribute or relationship's attribute, the operator (equal, less than, etc) and type in the value. Clicking on the "And", "Or" and "Not" buttons at the bottom of this view allows you to build a complex qualifier. By selecting the other tabs within this view, you can set the sort ordering, prefetching rules, return type (enterprise objects or raw rows), fetch limits, and other options related to this fetch specification. Additionally, using the last tab titled "SQL", you may type in the raw SQL to use for the fetch. Essentially, you can use the EOModeler graphical user interface to define all the details of a fetch specification instead of having to do so programmatically. This feature is especially useful for beginners and for frequently-used fetches. You may also access this feature by choosing the "Property" menu option and selecting "Add Fetch Specification".

The icon symbolized by a full disc, and eighth of a disc and an arrow flattens a property selected through a relationship into the "root" entity.

Other key menu items include:

Property-> Create Subclass (creates a new entity with the parent class already set)
Property-> Join in Many-to-Many (select two entities first, then select this option)
Property-> Generate Client Java Files (generates the Java Client files for an entity)

Tools:
Preferences (allows you to set the default behavior of the EOModeler application)

Conclusion

EOModeler is a powerful tool that allows you to quickly and easily define the mapping between the database and your custom enterprise objects. For more information on EOModeler and the Enterprise Objects Framework, consult the Enterprise Objects Framework Developer's Guide and the online book Enterprise Objects Framework Tools and Techniques (included in developer releases of Mac OS X Server and WebObjects).

Tips

- When selecting an attribute, don't click on the primary key, class property, used-for-locking, or not null columns or you will toggle the value as well as select the row. Click on the attribute or relationship name to select the row.

- You can change the default structure of the Objective-C or Java code generated by EOModeler by changing the h.template, m.template and java.template files in your Application directory.

Phone: 817-335-7770
E-mail: theresa@tensor.com
URL: http://www.tensor.com

Resources…

http://developer.apple.com/techpubs/webobjects
WebObjects Developer's Guide
Enterprise Objects Framework Developer's Guide
Enterprise Objects Framework Tools and Techniques (online with developer release)
http://www.omnigroup.com/MailArchive/WebObjects
http://www.omnigroup.com/MailArchive/eof
http://www2.stepwise.com/cgi-bin/WebObjects/Stepwise/Sites
ftp://dev.apple.com/devworld/Interactive_Media_Resources
http://www.apple.com/developer
http://developer.apple.com/media
http://til.info.apple.com/

About the Author…

Theresa Ray is a Senior Software Consultant for Tensor Information Systems in Fort Worth, TX (http://www.tensor.com). She has programmed in OPENSTEP (both WebObjects and AppKit interfaces) on projects for a wide variety of clients including the U.S. Navy, the United States Postal Service, America Online, and Lockheed-Martin.   Her experience spans all versions of WebObjects from 1.0 to 4.0, EOF 1.1 to 3.0, NEXTSTEP 3.1 to OPENSTEP 4.2, Rhapsody for Power Macintosh, and yellow-box for NT. In addition, she is an Apple-certified instructor for WebObjects courses.

Tensor Information Systems is an Apple partner providing systems integration and enterprise solutions to its customers. Tensor's employees are experienced in all Apple technologies including OPENSTEP, NEXTSTEP, Rhapsody, EOF and WebObjects.   Tensor also provides Apple-certified training in WebObjects, Oracle consulting and training, as well as systems integration consulting on HP-UX. And Oracle.

You may reach Theresa by e-mail: theresa@tensor.com or by phone at (817) 335-7770.

 Think different.

Interactive Media Resources Include:

Interactive Media Guidebooks

Market Research Reports

Survival Guides— Technical "How To" Guides

Comarketing Opportunities

Special Discounts

Apple Developer Connection

As Apple technologies such as QuickTime, ColorSync, and AppleScript continue to expand Macintosh as the tool of choice for content creators and interactive media authors, the Apple Developer Connection continues its commitment to provide creative professionals with the latest technical and marketing information and tools.

Interactive Media Resources

Whether looking for technical guides from industry experts or for market and industry research reports to help make critical business decisions, you'll find them on the Interactive Media Resources page.

Apple Developer Connection Programs and Products

ADC programs and products offer easy access to technical and business resources for anyone interested in developing for Apple platforms worldwide. Apple offers three levels of program participation serving developer needs .

Membership Programs

*Online Program—*Developers gain access to the latest technical documentation for Apple technologies as well as business resources and information through the Apple Developer Connection web site.

*Select Program—*Offers developers the convenience of technical and business information and resources on monthly CDs, provides access to prerelease software, and bundles two technical support incidents.

*Premier Program—*Meets the needs of developers who desire the most complete suite of products and services from Apple, including eight technical support incidents and discounts on Apple hardware.

Make the Connection.
Join ADC today!
http://developer.apple.com/ programs

Standalone Products

Apple offers many standalone products that allow developers to choose their own level of support from Apple or enhance their Select or Premier Program membership. Choose from the following products and begin enjoying the benefits today.

*Developer Connection Mailing—*Subscribe to the Apple Developer Connection Mailing for the latest in development tools, system software, and more.

*Technical Support—*Purchase technical support and work directly with Apple's Worldwide Developer Technical Support engineers.

*Apple Developer Connection News—*Stay connected to Apple and developer-specific news by subscribing to our free weekly e-mail newsletter, Apple Developer Connection News. Each newsletter contains up-to-date information on topics such as Mac OS, Interactive Media, Hardware, Apple News and Comarketing Opportunities.

Macintosh Products Guide

The most complete guide for Macintosh products! Be sure to list your hardware and software products in our *free* online database!

http://developer.apple.com/media
The ultimate source for creative professionals.