initWithContentsOfFile:	
	initWithPropertyList:
	initWithName:
Getting the filename path	
Getting a property list representation	
	modelAsPropertyList
Getting the name name	
Using entities addEntity:	
	removeEntityNamed:
	entityNamed:
	entities
Checking references referencesToProperty:	
Getting an object's entity entityForObject:	
Adding model information incorporateModel:	
Setting the adaptor bundle setAdaptorName:	
	adaptorName
Setting the connection dictionary	
	setConnectionDictionary:
	connectionDictionary
Setting the user dictionary setUserDictionary:	
	userDictionary

(NSString *)adaptorName

Returns the name of the adaptor for the model. This name can be used with EOAdaptor's adaptorV
method to create an adaptor.

(BOOL)addEntity:(EOEntity *)anEntity

Adds anEntity to the model. Returns YES if successful, NO if an entity with the same name alread

removeEntityNamed:

(NSDictionary *)connectionDictionary

Returns a dictionary containing information used to connect to the database server. The connectio
convenient place to specify default login information for applications using the model. See the EO
specification for more information.

(NSArray *)entities

Returns an array containing the EOModel's entities.

(EOEntity *)entityForObject:anEO

Returns the entity associated with anEO, whether anEO is an instance of an enterprise object class,
EOGenericRecord, or a fault object (see the EOFault class specification for information on faults).
has no associated entity.

(EOEntity *)entityNamed:(NSString *)name

Returns the entity named name, or nil if no such entity exists.

(BOOL)incorporateModel:(EOModel *)aModel

Copies the contents of aModel into the receiver. Returns YES if successful, NO if any identically n
two models differ, or if the two models use different adaptors. Doesn't affect the connection dictio

You can use this method to merge subsets of a larger model as they're loaded from individual reso

setConnectionDictionary:

initWithName:(NSString *)name

Initializes a newly allocated EOModel with name as its name. The EOModel needs to have entities
it's usable. This is the designated initializer for the EOModel class. Returns self.

initWithPropertyList:aPropertyList

Initializes a newly allocated EOModel from aPropertyList, which is a property list representation o
modelAsPropertyList to an existing model. Property list representations are used to save models to
files. initWithContentsOfFile: is roughly equivalent to this code excerpt:

modelAsPropertyList

Returns a string object encoding the EOModel as an ASCII property list. This representation can b
later reloaded using initWithContentsOfFile:. The following code excerpt saves an EOModel to a f
eomodel:

initWithPropertyList:

(NSString *)name

Returns the model's name.

(NSString *)path

Returns the full path of the model file used to create the EOModel (including the .eomodel extensi
wasn't initialized from a file.

(NSArray *)referencesToProperty:aProperty

Returns an array of all properties in the model that reference aProperty: derived attributes, relation
aProperty, and so on. aProperty itself may be either an EOAttribute or an EORelationship.

(void)setAdaptorName:(NSString *)adaptorName

Sets the name of the model's adaptor to adaptorName.

(void)setConnectionDictionary:(NSDictionary *)aDictionary

Sets the dictionary containing information used to connect to the database to aDictionary. Note tha
already been initialized with the model, this method does not propagate the connectionary down to
EOAdaptor class specification for more information on working with setConnectionDictionary:.

(void)setUserDictionary:(NSDictionary *)aDictionary

Sets the dictionary of auxiliary data, which your application can use for whatever it needs. This dic
property list (that is, it must contain only NSString, NSData, NSArray, and NSDictionary objects).

(NSDictionary *)userDictionary

Returns a dictionary of user data. Your application can use this to store any auxiliary information i
is a property list (that is, it contains only NSString, NSData, NSArray, and NSDictionary objects).