# Release Notes:
# Enterprise Objects Framework 2.2

This document contains the release notes for release 2.2 of Enterprise Objects Framework, included with release 3.5 of WebObjects.

Information about the release discovered since manufacture is continually being incorporated into the release notes, the most current version of which is posted on the Web at **http://ent.apple.com/NeXTanswers/HTMLFiles/2455.htmld/2455.html**.

The Windows version of this release can be used to develop software for Intel machines running Windows NT. The MachOS version of this release is 3-way fat; it can be used to develop software for NeXT, Intel, and SPARC. The PDO version of this release can be used to develop software for Solaris and HPUX.   None of the versions have been qualified for OpenStep application development.

Note that the installation of WebObjects 3.5 replaces your existing OpenStep installation with new development tools, frameworks, and files.   For example, there are modificiations to ProjectBuilder, makefiles, and project types to accomodate new features in WebObjects. Interface Builder and the AppKit framework are installed with WebObjects 3.5, but have not updated from the OPENSTEP 4.2 release.   While we don't believe there are problems developing OpenStep applications on a WebObjects 3.5 system, you will not receive support related to problems you encounter while developing OpenStep-based applications.   If you need to develop OpenStep applications on a supported platform, use a machine that does not have WebObjects 3.5 installed.

These release notes are organized into the following sections:

· Information About Other Releases
· Notes specific to Windows NT Developers

· Notes specific to PDO Developers
· Database Client Libraries
· New Features in Release 2.2
· Locating the Documentation for this Release
· Installing the Examples
· Known Problems in Release 2.2
· Enterprise Objects Framework User Defaults

**Note:** Because release 2.0 of Enterprise Objects Framework and later releases are based on the same architecture, many of the subjects in this document are relevant to all 2.x releases. Consequently, the term "release 2.x" is used where a discussion applies equally to releases 2.0 and higher.

## Information About Other Releases

This document describes the features introduced, the bugs fixed, and the known problems in Enterprise Objects Framework release 2.2.

For a description of the features introduced in Enterprise Objects Framework release 2.0, see the 2.0 release notes. They contain a lot of information that's still relevant for 2.2, including information about converting release 1.x applications to 2.0 and a list of features shared with Enterprise Objects Framework release 1.2. The 2.0 release notes are available on the Web at **http://ent.apple.com/NeXTanswers/HTMLFiles/2455.htmld/2455.html**.

For a description of how Enterprise Objects Framework 2.x differs from the preceding releases, see the document *Differences Between Enterprise Objects Framework 1x and 2.0* in **/NextLibrary/Documentation/NextDev/EnterpriseObjects/1x_To_2**.

## Notes Specific to Windows NT Developers

References to filenames in this document use UNIX format. On Windows NT, you can interpret these by reading backslashes for the slashes and adding the installation directory (**c:\NeXT\** by default) to the beginning. For example, **/NextLibrary/Documentation** on

Windows NT is **c:\NeXT\NextLibrary\Documentation**.

To use Enterprise Objects Framework on Windows NT, you must have the appropriate database client libraries. The Sybase client libraries are provided on the WebObjects Enterprise 4.2 CD as an optional package. To install the Sybase client libraries, you must do a custom installation and explicitly specify that you want to install the package. To use Enterprise Objects Framework with Oracle or Informix, you must purchase the appropriate client libraries from your database vendor.

Here's what you need:

### Oracle

Phone: (800) 542-1170 or call your local sales representative
Ask for: SQLNET v2.2 for PC/Windows NT

The Oracle adaptor on NT requires the Oracle 8.0, 7.3, or 7.2 Client Library. It won't work with the 7.1 libraries.

### Informix

Phone: (800) 331-1763 or call your local sales representative
Ask for: ESQL/C version 7.2 for Win32

## Notes Specific to PDO Developers

To use Enterprise Objects Framework on PDO, you must have the appropriate database client libraries.

Here's what you need:

### Oracle

Phone: (800) 542-1170 or call your local sales representative
Ask for: 7.3.2 SQLNet V2 TCP/IP Client libraries

### Informix

Phone: (800) 331-1763 or call your local sales representative

Ask for: ESQL/C Version 7.20.UC2

**Sybase**

Phone: (800) 685-8225 or call your local sales representative
Ask for: OpenClient/C Version 11.1

On PDO applications must explicitly link against the adaptor framework and the client libraries. New makefiles look for adaptor frameworks and automatically add in the right linker arguments. Simply add the adaptor framework to your project, and set the requisite environment variable specifying where the client libraries are installed. For Oracle set ORACLE_HOME and optionally ORACLE_REL. (The ORACLE_REL flag controls which set of libraries are used. It uses the Oracle 7.3 static link libraries by default, but you can also specify "8.0-static" or "7.3-dynamic.") For Sybase set SYBASE_HOME. For Informix set INFORMIX_HOME.

If you use dynamic libraries on Solaris, you need to set the LD_LIBRARY_PATH environment variable when running your application.

## Database Client Libraries

This section includes some tips on using database client libraries with Enterprise Objects Frameworks. It is organized by database vendor.

**Oracle**

On Windows NT, using the latest release of the client library (7.3) requires you to use SQL*Net v2, which requires a **tnsnames.ora** file. **tnsnames.ora** is a file that you put on client machines, generally in the directory **Orant/Network/Admin**. The file contains information needed to connect to a server over the network. Entries in **tnsnames.ora** are keyed off of a server ID alias, and they include information such as the server ID, the host machine name, and the network protocol used by the client library to resolve the server ID alias. An entry in **tnsnames.ora** might resemble the following:

```
myServerAlias = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
   (HOST=myMachine) (PORT=1521))(CONNECT_DATA=(SID=eof)))
```

Oracle provides tools you can use to create **tnsnames.ora** files. Refer to your Oracle documentation for more information on **tnsnames.ora** files and the tools you can use to create them.

If you're using the 7.2 version of the Oracle client libraries on Windows NT or if you're using Mach clients, you can use either SQL*Net v1 or SQL*Net v2. To use SQL*Net v1, you should set your adaptor's connectionDictionary **serverId** entry to "T:<host-machine>:<server-name>".

To use SQL*Net V2 on Mach clients, you should create a **tnsnames.ora** file and put it in the **/etc** directory.

### Informix

If you get the error "INFORMIXSERVER not in sqlhosts file (25596)" but can connect to your database server using the Informix ilogin program, you may need to run SetNet32 to update the environment variables used by Informix.

The Informix client libraries appear to have redundant sources of server information. They use the **sqlhosts** file (**$INFORMIXDIR/etc/sqlhosts**) as well as a collection of environment variables managed by the Setnet32 program.

See your Informix documentation for more information on the sqlhosts file and the Setnet32 program.

## New Features for 2.2

### Java-wrapped Access and Control Layers

EOF 2.2 includes:

- Java-wrapped versions of the EOAccess and EOControl frameworks, so that you can write your Enterprise Object classes and Enterprise Objects Framework code in Java.
- A Java class browser demo application for browsing Java APIs, available in **NextDemos/JavaBrowser.app**.

### EOModeler

- EOModeler can generate template code for Java enterprise object classes.
- Reverse engineering incorporates more schema information from your database into newly created models.

### Framework

EOF 2.2 provides:

- Caching of to-many array faults that greatly reduces faulting (especially in WebObjects applications), thereby improving performance.
- A new insensitive like qualifier.

## Locating the Documentation for this Release

The documentation for this release is provided in HTML and PDF formats. You can locate and view the documentation using the WebObjects Home Page. On NT, click the Start button and choose WOHomePage in the WebObjects program group. Or open the home page in a web browser directly, using the file <*DocRoot*>**/WebObjects/Documentation/WOHomePage.html**.

In addition to the documentation listed in the home page, this release also includes the document *Differences Between Enterprise Objects Framework 1x and 2.0* on-line, which describes how the product has changed between the 1x and 2.0 releases. It is located in the directory: **/NextLibrary/Documentation/NextDev/EnterpriseObjects/1x_To_2**.

### Problems with Reading PDF Files

If you're using the Acrobat Reader to view PDF files on Windows NT, some text may initially appear as gray lines. If this happens, you can fix it by choosing File -> Preferences -> General. In the General Preferences panel, uncheck the Greek Text option.

### Documentation Feedback

Your comments on our documentation are especially valuable. Please send electronic mail with your comments and suggestions to **techpubsfeedback@group.apple.com**.

# Installing the Examples

This release provides on-line examples to help familiarize you with Enterprise Objects Framework 2.x. These examples are located in /**NextDeveloper/Examples/EnterpriseObjects**.   For more information on installing and building the examples, see the *ExamplesGuide* within the examples directory or refer to *Post Installation Instructions*, which is accessible from the WebObjects Home Page.

# Known Problems in Release 2.2

This section describes known problems with release 2.2 of Enterprise Objects Framework. It is organized into the following sections:

- Java-wrapped Access and Control Layers
- Access Layer
- Control Layer
- Miscellaneous Framework
- Informix Adaptor
- ODBC Adaptor
- Oracle Adaptor
- Sybase Adaptor
- EOModeler
- On-line Examples
- Documentation

### Java-wrapped Access and Control Layers

These problems exist in the Java-wrapped versions of the Access and Control layers of this Enterprise Objects Framework release:

Reference: 81432

Problem: Key-value coding invocations raise if the provided key is different in Java from what it is in Objective-C.

Description: If you use key-value coding to invoke a method whose name is different in Java from what it is in Objective-C (for example, ImmutableVector.size is mapped to [NSArray count]), the key is *unbound* on the Objective-C side and an exception is raised.

Workaround: In key-value coding invocations, use the Objective-C keys instead of their Java counterparts.

Reference: 81711

Problem: **valueForKey** raises an unbound key exception if the Objective-C method corresponding to the provided key returns **void**.

Workaround: If you're writing Objective-C code that you're going to bridge, write Objective-C methods that return a value (instead of **void**).

### Access Layer

These problems exist in the access layer of this Enterprise Objects Framework release:

Reference: 81863

Problem: There isn't a publicly available method to turn off to-many array fault caching.

Workaround: If you need to turn off to-many array fault caching for any reason (if you've implemented your own solution, for example), invoke the following method before the creation of any instances of EODatabase or EODatabaseContext. Calling it from main() would be good.

```
[EODatabaseContext _setUseToManyCaching:NO];
```

Reference: 82008

Problem: NSNumber with value 0 is still taken to be NULL for primary key.

Description: Starting in EOF 2.2, the EODatabaseContext assumes that an object with a single attribute primary key with a value of zero must be a newly created instance. And as such, the database context will attempt to get a new primary key for the object via delegate hook or the adaptor-specific primary key generation mechanism. This change allows users to use scalar data types (such as **int**) as an enterprise object's primary key, and still rely on EOF automatic primary key generation.

Unfortunately, if you have an existing database containing rows that have a primary key of 0, an attempt to update an object created from such a row will cause the database context to incorrectly assume that an object created from such a row needs to get a new primary key. This can leave invalid foreign key references in other tables.

Workaround: Implement the EODatabaseContext delegate method **databaseContext:newPrimaryKeyForObject:entity:** to catch the case where the framework is going to mistakenly get a new key.

From EODatabaseContext.h:

```
- (NSDictionary *)databaseContext:(EODatabaseContext *)context
    newPrimaryKeyForObject:(id)object
    entity:(EOEntity *)entity;
    // If a newly inserted EO doesn't already have have a primary key set,
    // this delegate is called to generate a key.   If the delegate is not implemented,
    // or returns nil, then the DatabaseContext will call
    // [EOAdaptorChannel primaryKeyForNewRowWithEntity:(EOEntity *)entity]
    // to attempt to generate the key.
```

Example of how to get the database context and set its delegate:

```
{
...
    model = [[EOModelGroup defaultGroup] modelNamed:@"myModel"];
    dbContext = [EODatabaseContext
        registeredDatabaseContextForModel:model
        editingContext:editingContext];
    [dbContext setDelegate:self];
...
}
```

Example of a delegate method implementation:

```
- (NSDictionary *)databaseContext:(EODatabaseContext *)context
    newPrimaryKeyForObject:(id)object
    entity:(EOEntity *)entity
{
    if (entity == entityThatIKnowHasAValidRowWithAKeyOfZero)
```

```
        if ([[object valueForKey:primaryKeyThatCanBeZero] isEqual:[NSNumber zero]])
            return [NSNumber zero];
    return nil;
}
```

Reference: 77721

Problem: Can't update rows containing string attributes with trailing spaces.

Description: The Enterprise Objects Framework adaptors automatically strip trailing spaces from string values fetched from the database. They also strip spaces on newly inserted strings before sending them to the database. This works fine until you fetch data that a non-Enterprise Objects Framework application inserted that contains trailing spaces. If that attribute is marked as "used for locking," Enterprise Objects Framework will be unable to update the row (because the version in the database is different from the space-stripped version in the snapshot).

Workaround: Do not mark columns that could contain trailing spaces as used for locking. Alternately, strip the spaces from the data in your server.

Reference: 77546

Problem: Inclusion of framework models in defaultModelGroup doesn't happen automatically on PDO.

Description: Applications on PDO that link with frameworks that include models don't automatically get the models put in the applications.

Workaround: Manually construct the the model group yourself at application initialization. For example:

```
    EOModelGroup *group = [EOModelGroup new];

// repeat for each model containing framework used by app
    NSBundle *bundle = [NSBundle bundleForClass:[SomeClassInFramework class]]
    NSEnumerator *enumerator = [[bundle pathsForResourcesOfType:@"eomodeld"   inDirectory:nil] objectEnumerator];
    NSString *modelPath;
    while (modelPath = [enumerator nextObject])
        [group addModelWithFile:modelPath];

    [EOModelGroup setDefaultGroup:group];
```

```
[group release];
```

Reference: 77631

Problem: If you're editing a model (in code or with EOModeler), sometimes an EORelationship will return **nil** from **inverseRelationship**, even though you just added the inverse relationship.

Description: The first time you ask an EORelationship for its inverse relationship, it searches all the relationships in its **destinationEntity** looking for an inverse. It caches the result of this search, and this cache does not always get invalidated when the relationships of the **destinationEntity** change.

Workaround: See the category on EORelationship in the **ModelerBundle/RelationshipExtras.m** example.

Reference: 77354

Problem: Problems saving changes to attributes with mutable custom value objects.

Description: Suppose you have a mutable custom value type, PhoneNumber, that implements methods such as **setAreaCode:** and **setPrefix:** to change an enterprise object's values. If you use such methods to modify an enterprise object's values and save changes, the enterprise object is not saved to the database.

Workaround: To use mutable value classes you must do three things:

1. In the PhoneNumber value class object, implement **isEqual:** to appropriately compare two instances.

2. If the PhoneNumber value is about to be modified, its owning enterprise object must invoke `[self willChange]` before the modification.

3. In your enterprise object class, you must implement a "set" method that copies rather than retains the object passed to it. For example:

```
- (void)setDayTimePhone:(PhoneNumber *)number {
    [dayTimePhone autorelease]
```

```
        dayTimePhone = [number copy];
    }
```

#1 would be required, even in a perfect world. #2 is a requirement of the basic Enterprise Objects Framework architecture. #3 is a bug in Enterprise Objects Framework -- Enterprise Objects Framework should be passing enterprise objects a copy of their values, but instead it's passing the the same instance that's shared in the snapshot.

In general, the simplest workaround is to use **immutable** custom value objects.

Reference: 72619

Problem: EOAdaptorChannel does not inherit EOAdaptor's delegate.

Description: According to the EOAdaptorChannel class specification and header file, the EOAdaptorChannel's delegate should automatically be kept in synch with its adaptor. Indeed, if a new delegate is assigned to an EOAdaptor, the new delegate is propagated to any of that adaptor's existing contexts and channels. However, if you create an EOAdaptor, assign a delegate to it, and then create EOAdaptorContexts and EOAdaptorChannels, the newly-created contexts and channels don't have the delegate assigned to them.

Workaround: Wait until all of the channels and contexts are instantiated before assigning the adaptor's delegate, or reassign the adaptor's delegate each time a new channel or context is created.

Reference: 61475

Problem: Enterprise Objects Framework performs less efficient deep fetches for single-table inheritance mappings than it should.

Description: To perform a deep fetch, Enterprise Objects Framework performs a fetch for each concrete class in an inheritance hierarchy. For a single-table inheritance mapping, Enterprise Objects Framework should perform only one fetch and then sort the results in memory. For example, if Person, Employee, and Customer objects are stored in one tableÐthe PERSON tableÐthe Framework should perform one fetch on the PERSON table to fulfill a deep fetch request of Person, Employee, and Customer objects. Instead, it performs three fetches: one to get Person objects, one to get Employee objects, and another to get Customer objects.

Workaround: Define only one entity for the entire inheritance hierarchy, and use the EOModelGroup delegate methods **subEntityForEntity:primaryKey:isFinal:**, and **entity:classForObjectWithGlobalID:** to create instances of the proper subclasses from database rows.

Reference: 74251

Problem: Changes made during **saveChanges** are silently lost.

Description: If you change an enterprise object while its editing context is in its **saveChanges** method (for example, if you change an enterprise object in an EODatabaseContext delegate method), the changes may be silently lost.

Workaround: Don't make changes to objects during the save process. Instead, make the changes from the EOEditingContext delegate method **editingContextWillSaveChanges:**.

Reference: 74345

Problem: You can't update to-many relationship if the foreign key isn't marked as a class property or as used for locking.

Description: Suppose that an entity's foreign key attribute isn't marked as a class property or as used for locking. If you designate the foreign key attribute as a to-many relationship's destination key, the foreign key value isn't always updated. This occurs because the destination entity doesn't know that the attribute participates in a relationship. Therefore, the destination entity doesn't fetch the foreign key from the database or update it.

Workaround: Mark attributes that are destination keys of a to-many relationship so they are fetched. For example, you could:

- Mark them as class properties.
- Mark them as used for locking.
- Use them in inverse relationships to the problematic to-many's source entity.

Reference: 74379

Problem: EODatabaseContext can't be the direct parentObjectStore of an EOEditingContext.

Description: The EODatabaseContext requires that an EOObjectStoreCoordinator sit between it and any EOEditingContexts that it serves. This is the default configuration set up by the framework, so you shouldn't normally run into this problem. Just a reminder, you can set up an editingContext and be ready to go with this one line:

```
EOEditingContext *editingContext = [EOEditingContext new];
        // automatically uses [EOObjectStoreCoordinator defaultCoordinator]
        // as parentObjectStore
```

Any necessary EODatabaseContexts are created and registered automatically.

Workaround: Don't assign an EODatabaseContext as the parentObjectStore of an EOEditingContext. There's no benefit to doing so anyway.

Reference: 76526

Problem: Applying a qualifier with key path to top of horizontally mapped inheritance hierarchy generates invalid SQL.

Description: Enterprise Objects Framework's query building mechanism doesn't handle relationships to inheritance hierarchies. For example, suppose that you are are attempting to qualify a fetch through a to-many relationship (**planes**) that points to the top of a horizontally mapped inheritance hierarchy (for the entities Plane, FighterPlane, and TrainerPlane). If you want the query to test against all tables, you'd expect Enterprise Objects Framework to generate SQL similar to the following:

```
SELECT  t0.AIRPORT_ID
FROM PLANE t1, FIGHTER t2, TRAINER t3, AIRPORT t0
WHERE
    (t1.LENGTH <= 1000 AND t0.AIRPORT_ID  = t1.AIRPORT_FK) OR
    (t2.LENGTH <= 1000 AND t0.AIRPORT_ID  = t1.AIRPORT_FK) OR
    (t3.LENGTH <= 1000 AND t0.AIRPORT_ID  = t1.AIRPORT_FK)
```

Instead, Enterprise Objects Framework generates the following SQL:

```
SELECT  t0.AIRPORT_ID
FROM PLANE t1, AIRPORT t0
WHERE
    (t1.LENGTH <= 1000) AND
    t0.AIRPORT_ID = t1.AIRPORT_FK
```

In other words, only the table for the root of the hierarchy is queried.

Workaround: You can create a qualifier that generates the correct SQL by:

1. Adding relationships in the source entity to all the tables in the inheritance hierarchy. For example, to the Airport entity, you'd add the relationships **toFighters** and **toTrainers** to the destination entities FighterPlane and TrainerPlane, respectively. Mark the relationships so they aren't class properties.
2. When building your query, explicitly list these extra relationships. In the Planes example, you'd fetch from Airport where "planes.length < 1000 OR toFigtherPlanes.length < 1000 OR toTrainerPlanes.length < 1000"

Alternatively, you might be able to solve this problem more generally by writing a post processor for EOQualifiers that splits up clauses that perform inheritance tests. The post processor could even programmatically generate the additional relationships on demand and register them with the model using names like "plane_Subclass_Fighter".

A generic EOQualifier post processor could be wired into Enterprise Objects Framework so that application writers don't have to know it exists. The right place for such a mechanism is probably in EOKeyValueQualifier's **schemaBasedQualifierWithRootEntity:** method (see EOSQLQualifier.h). You could put the post processor code in a subclass of EOKeyValueQualifier (with an appropriate call to **super** after the transformation, if any, is performed) and have your subclass pose as EOKeyValueQualifier.

Reference: 47832

Problem: Enterprise Objects Framework can't update attributes whose internal types are custom (such as NSImages).

Description: A custom value class must implement **isEqual:** to be used for attributes marked as used for locking.

Workaround: Implement **isEqual:** in the custom value class or don't mark the attribute as used for locking.

Reference: 65078

Problem: Seemingly innocuous qualified fetch always causes exception.

Description: Some qualified fetches raise exceptions because values in the SQL have been formatted as strings when they should have been formatted as some other type. This can happen when you enter an invalid external type or when a Sybase model doesn't contain information about user-defined types that are used in the model.

Workaround: In the case of an invalid external type, simply correct it. In the case of a user defined type, create a new model by reverse engineering the database. The new model's connection dictionary contains information about user-defined types. Copy the connection dictionary from the new model to the original one.

Reference: 69039

Problem: Alert panel displaying adaptor error is never dismissed.

Description: This occurs whenever an adaptor operation that was invoked from an **awakeFromNib:** method displays an alert panel. The problem is that the entire object graph is not yet instantiated when **awakeFromNib:** is invoked.

Workaround: Database operations should be begun from the **applicationDidFinishLaunching:** method rather than the **awakeFromNib:** method. Of course, this means that any methods that indirectly cause database communication should also be invoked from the **applicationDidFinishLaunching:** method.

Reference: 76885

Problem: The set of valid values for the **databaseEncoding** entry of a connection dictionary are not documented, and the set can vary in different locales.

Description: A connection dictionary's **databaseEncoding** entry contains the localized name for the string encoding. The localized names are not documented. Furthermore, the localized names for string encodings can vary with the user's locale. Consequently, the specified encoding for an application might not work for users in different locales.

Workaround: You can find the localized name for string encodings in **/NextLibrary/Frameworks/Foundation.framework/Resources/*language*.lproj/EncodingNames.strings**.

Reference: 59472

Problem: Derived attributes are limited and don't offer full SQL as advertised.

Description: Placing a string or a numeric constant in the definition field of a derived attribute generates invalid SQL. Definitions such as ªtitleº and ª0.0º don't work correctly. However, definitions such as ªatt1 + 5º should work correctly when ªatt1º specifies another attribute.

Workaround: None.

Reference: 70049

Problem: EOModelGroup doesn't raise an exception when more than one entity has the same name.

Description: Although it is illegal to have the same entity name in two different models in a model group, EOModelGroup doesn't check to see if this is the case when adding a model.

Workaround: Manually verify that no two models have entities with the same name or write a method to perform the check.

Reference: 70260

Problem: Some error messages returned by the Framework's default validation methods (such as methods that check that a value doesn't exceed the maximum width specified in its attribute) aren't localized.

Workaround: Implement an exception handler for use with the EOEditingContext, and have the handler replace the error messages with localized strings.

Reference: 76152

Problem: Inserting and deleting objects involved in inverse, to-many relationships can be very slow.

Description: Suppose that a ServiceRequest has a to-one relationship to its CustomerServiceRepresentative, and that the CustomerServiceRepresentative has an inverse to-many relationship to its ServiceRequests. When you assign a request to a representative using **addObject:toBothSidesOfRelationshipWithKey:**, you fire the fault for the **CustomerServiceRepresentative.serviceRequests**. So, if a representative has a large number of requests, assigning a new request to a representative can be very slow. Correspondingly, when you delete a request, Enterprise Objects Framework fires the corresponding representative's **serviceRequests** fault so it can remove the request from the array.

Workaround: Set the inverse, to-many relationship so it isn't a class property. For example, in the above request-representative scenario, you would remove the **serviceRequests** relationship from the class properties of the CustomerServerRepresentative entity.

Reference: 82173

Problem: EOAdaptorChannel.h has an erroneous comment for the method **fetchRowWithZone:**.

Description: When **fetchRowWithZone:** returns **nil** signalling the end of the result set, **isFetchInProgress** can return **YES** to indicate that there are more result sets. The comment should read:

```
- (NSMutableDictionary *)fetchRowWithZone:(NSZone *)zone;
    // Fetches the next row from the result set of the last
    // -selectAttributes:...   message and returns values for the attribute
    // names in attributes. When no rows are left, this method invokes
    // adaptorChannel:didFinishFetching:, and returns nil.
    // For adaptors that can have multiple result sets (ODBC and Sybase),
    // This method will return nil at the end of each result set, however
    // isFetchInProgress will return YES. The channel will also send the
    // delegate an an adaptorChannelDidChangeResultSet: message whenever
    // the end of one result set has been reached and another one is
    // pending.
    // This methods may raise an exception if an error occurs.
```

### Control Layer

These problems exist in the control layer of this Enterprise Objects Framework release:

Reference: 82210

Problem: Attempt to copy an EOSortOrdering object throws an exception:

```
*** -[EOSortOrdering copyWithZone:]: selector not recognized
```

Workaround: Don't copy EOSortOrderings, or implement **copyWithZone:** in a category on EOSortOrdering.

Reference: 68146

Problem: Fault failure leaves EOInterface layer unstable.

Description: If a displayed enterprise object has a to-one relationship to a non-existent destination row, attempting to access the destination object raises an exception and corrupts the state of user interface objects. (Enterprise Objects Framework raises an exception when a to-one relationship cannot be resolved due to a referential integrity problem in the database).

Workaround: See the chapter "Advanced Enterprise Object Modeling" in the *Enterprise Objects Framework Developer's Guide* for information on handling optional to-one relationships.

Reference: 64084

Problem: Inserted enterprise objects don't get removed from EODisplayGroup after sending **revert** to EOEditingContext.

Description: If you fetch objects into an EODisplayGroup, insert a few objects, delete a few objects, and then update a few objects; telling the EOEditingContext to revert backs out the updates, but not insertions or deletions. The actual insertions and deletions have been reverted, but the EODisplayGroup doesn't know how to revert its object list (since is doesn't keep track of what its original object list was before the insertions and deletions).

Workaround: Programmatically tell all affected EODisplayGroups to refetch after telling the EOEditingContext to revert, or use **refetch:** (**invalidateAllObjects:**) instead of revert.

Reference: 72177

Problem: Deleted objects are still registered in the EOEditingContext after a **saveChanges** operation.

Description: If an object is deleted in the EOEditingContext and then the EOEditingContext is saved, the deleted object isn't forgotten by the EOEditingContext (i.e., sending the EOEditingContext the message **objectForGlobalID:** should return **nil**, but it doesn't).

Workaround: Fortunately, this should have no affect on most applications. However, if you need to work around this problem, remember the deleted objects in the **editingContextWillSaveChanges:** delegate, and then invoke **forgetObject:** for each object after a successful save.

Reference: 72269

Problem: You can't tell Enterprise Objects Framework to not undo your changes when delete propagation fails.

Description: The EOEditingContext delegate method **editingContextShouldUndoUserActionsAfterFailure:** is supposed to allow programs to indicate that they do not want user actions undone if a validation or delete propagation error occurs. However, the EOEditingContext undoes the user action regardless of the return value from the method.

Workaround: None.

Reference: 72903

Problem: Aborted deletions are incorrectly recorded in the undo stack as empty undos.

Description: When the deletion of an object fails due to a deny rule, the undo manager records an empty undo group. This means that although nothing actually happened during the operation, the operation still needs to be undone before previous operations can be undone. For interactive programs this is not normally a problem since users rely on visual cues to determine how many times to undo. However, this may be a problem for code that programmatically performs undo operations.

Workaround: None.

Reference: 74965

Problem: The EOEditingContext delegate method **editingContext:shouldPresentException:** isn't called when **saveChanges** is invoked programmatically.

Description: If you use the following code:

```
eo = [[[Movie alloc] init] autorelease];
[eo setTitle:nil];     // Assume that the TITLE column doesn't allow NULLs
[editingContext insertObject:eo];

[editingContext saveChanges];
```

the **editingContext:shouldPresentException:** delegate method isn't called. Instead, an exception is raised.

However, if you instead set a user interface control object as the target for the editing context's **saveChanges:** action, the delegate method is invoked.

Workaround: Always use **saveChanges:** instead of **saveChanges**.   If you don't have an argument to pass, just use

```
[editingContext saveChanges:nil];
```

Reference: 76466

Problem: EOEditingContexts don't propagate uninserts.

Description: Propagate delete works fine for an object that that is fetched from the database. However, if an object is created and inserted in an editing context and then deleted (thus never being saved to the database), the delete isn't propagated to the destinations of the object's relationships. For example, suppose you create an ExpenseReport object and several LineItems for it. If you then delete the ExpenseReport, the delete doesn't propagate to the LineItems. If the LineItems can't exist without an owning ExpenseReport, EOEditingContext's **saveChanges** fails when the delete is performed within the same event as the insert.

Workaround: Call [editingContext processRecentChanges] before deleting the previously inserted object.

Reference: 76901

Problem: Associations are refreshed too early when a modal panel (such as the fetch limit panel) is raised, possibly causing an exception.

Description: Suppose you have two EODisplayGroups that are set to fetch on load. If the second display group to fetch exceeds the fetch limit, an alert panel inquiring whether to continue the fetch pops up. Presenting the panel causes the run loop to flush the first display group's user interface drawing. If the first display group's user interface displays enterprise object properties via key paths, EOFaults may be fired. Since the default EODatabaseChannel is busy with the fetch for the second display group, an exception is raised because there isn't an available channel.

Workaround: There are several workarounds:

1. Don't use the fetch "prompt on limit" feature.
2. If your application only uses modal panels for alerts and warnings, disable the modal refresh by sending the following message in your application's initialization code (make sure to awake the EODisplayGroup first so it's initialize method is called before you send this message):

   ```
   [[EODelayedObserverQueue defaultObserverQueue]
       setRunLoopModes:[NSArray arrayWithObject:NSDefaultRunLoopMode]]
   ```

3. Implement your own fetch limit panel that doesn't invoke a modal event loop.
4. Register an additional EODatabaseChannel with the EODatabaseContext so the fetch of the faults can occur while the second EODisplayGroup fetch is paused.

Reference: 77181

Problem: The EOQualifier contains operator doesn't do SQL generation.

Description: The contains operator is meant to do to-many array comparison. For example, you might query on the Studio entity using the following qualifier format:

```
"movies contains %@", someMovieObj
```

The resulting qualifier works for an in-memory search, but doesn't generate SQL for a database search.

On the other hand, using the equality operator for to-many array comparison kind of works in SQL, but not in memory. For example, you might query on the Studio entity using the following qualifier format:

```
"movies = %@", someMovieObj
```

The resulting qualifier returns no matches when evaluated in memory because the array property is not equal to the movie object as determined by **isEqual:**. However, when it's evaluated in the database, it returns the same studio N times (where N is the number of movies in the Studio's movies array), even though the qualifier is attempting to match only one movie.

Workaround: Use the contains operator for qualifiers in-memory evaluation and the equality (=) operator (with **usesDistinct** on the fetch specification) for evaluation in the database.

### Miscellaneous Framework

These problems exist in this Enterprise Objects Framework release:

Reference: 46679

Problem: Enterprise Objects Framework's private instance variables aren't declared **@private** and don't have names that begin with the underbar ('_') character.

Description: All instance variables in Enterprise Objects Framework should be considered private.

Workaround: Don't directly access Enterprise Objects Framework instance variables.

Reference: 69211

Problem: Link errors on Windows NT.

Description: Programs on Windows NT must add explicit references to at least one class in each framework in order to avoid link errors at run time.

Workaround: Add a function like that in the following code snippet, which refers to classes in each of Enterprise Objects Framework's layers. Though never invoked, it forces the appropriate linking to occur.

```
#ifdef WIN32
#import <EOControl/EOControl.h>
#import <EOAccess/EOAccess.h>
#import <EOInterface/EOInterface.h>

void _referenceAllEOFrameworks()
{
    [EODisplayGroup new];          // EOInterface
    [EOEntity new];                // EOAccess
    [EOEditingContext new];        // EOControl
}
#endif
```

If you create your project with the type "EOApplication," this code is automatically added to your project main file.

Reference: 72027

Problem: EOKeyValueCoding is broken for doubles on HPUX PDO.

Description: EOKeyValueCoding doesn't work correctly for accessor methods that set and return doubles on HPUX under PDO 4.1 (and previous releases). This means that users cannot correctly fetch enterprise objects that have class properties with accessor methods that use doubles.

Workaround: Change your enterprise objects' accessor methods to return and take as arguments NSNumbers, NSDecimalNumbers, or **int**s.

### Informix Adaptor

These problems exist in the Informix adaptor supplied with this release of Enterprise Objects Framework:

Reference: 64031

Problem: Informix adaptor user defaults for Enterprise Objects Framework 2.x are different than those for Enterprise Objects Framework 1.2.

Description: In Enterprise Objects Framework 1.2, the InformixAdaptor stores defaults in the EOFInformixAdaptor domain and uses the following keys: INFORMIXDIR, DBDATE, DBLANG, DBMONEY, InformixLogErrors, ShowSystables, ShowTableOwner, Beautify, DefaultIsolationLevel, GlobalOptimization, GlobalExplain, GlobalLockMode, GlobalPDQPriority, GlobalDataSkip, GlobalConstraints, DatabaseExclusive

Enterprise Objects Framework 2.x stores defaults in the standard NSGlobalDomain and prefixes all keys with ªInformixº, that is: InformixINFORMIXDIR, InformixDBDATE, InformixDBLANG, InformixDBMONEY, InformixLogErrors, InformixShowSystables, InformixShowTableOwner, InformixBeautify, InformixDefaultIsolationLevel, InformixGlobalOptimization, InformixGlobalExplain, InformixGlobalLockMode, InformixGlobalPDQPriority, InformixGlobalDataSkip, InformixGlobalConstraints, InformixDatabaseExclusive.

Workaround: None.

Reference: 70232

Problem: Informix adaptor raises an exception when you try to sort on attributes that are not in the select list.

Description: Due to a restriction in the Informix adaptor, it's not possible to sort on attributes that aren't included in the select list. This means that it isn't possible to sort the results using an attribute that is not marked on the entity as either a primary key, used for locking, or a class property.

Workaround: If possible, add the attribute to the entity and mark it as used for locking. Otherwise, there is no workaround.

### ODBC Adaptor

These problems exist in the ODBC adaptor supplied with this release of Enterprise Objects Framework:

Reference: 81774

Problem: Can't save large BLOB's to SQLServer

Description: If you try to save large BLOB's (more than 1 meg), you will probably encounter a segmentation violation. This is happening because something in the ODBC stack can't handle large buffers.

Workaround: You can add a special EOF flag to the **driverInfo** dictionary that will cause EOF to use the **SQLPutData()** function to break up large BLOB's into smaller pieces for transmission. To enable this behavior, add the following line to your **driverInfo** dictionary in the EOModel.

```
EOF_ENABLE_SQLPUTDATA = Y;
```

Starting with EOF 2.2, this flag is always set when you reverse engineer a model, but older models created with EOF 2.1 or EOF 2.0 will not have the flag.

**Warning to users of MS-Access:** The ODBC Access driver doesn't handle the **SQLPutData()** calls correctly, so Access users should not turn this flag on. It will be set to "N" by default when reverse engineering an Access database.

### Oracle Adaptor

These problems exist in the Oracle adaptor supplied with this release of Enterprise Objects Framework:

Reference: 82299

Problem: On NT, you can't use gdb on an app running against the oracle adaptor.

Description: It has been reported that running gdb on an application running against the OracleAdaptor can sometimes result in unexplained crashes or hangs, even when the same application works when run from outside the debugger. It seems that something in the Oracle Tracing mechanism causes the entire process to behave unpredictably when running within gdb. Some people never see this problem, others may see it occasionally, some may never be able to run gdb acceptably.

Workaround: There are several ways to deactivate this tracing. The easiest is to remove tracing from the registry by renaming the HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OTRACE73 entry.

Reference: 77517

Problem: If you try to update an enterprise object with an attribute that maps to a LONG RAW column while using on-demand locking, you get the following exception:

```
         "fetchObject -- EODatabaseChannel 0x12345678: attempt to
lock object that has out of date snapshot"
```

Description: It appears that the Oracle database sometimes returns wrong BLOB values when it's passed "SELECT ... <long-raw-column,... FOR UPDATE". So when the EODatabaseChannel attempts to acquire a lock on the row, the results of the SELECT don't match the results that were gotten on the last SELECT (without the FOR UPDATE clause).

Workaround: None. You can't use on-demand locking if you're going to be updating tables with LONG RAW columns.

Reference: 77366

Problem: If you're using pessimistic locking in combination with batch faulting or prefetching of relationships, you'll get the error "ORA-01786: FOR UPDATE of this query expression is not allowed" when you run your application.

Description: When you use pessimistic locking mode in combination with batch faulting or prefetching of relationships, Enterprise Objects Framework generates a SQL statement like the following:

```
<OracleSQLExpression: "SELECT DISTINCT   t0.CATEGORY, t0.DATE_RELEASED, t0.LANGUAGE, t0.MOVIE_ID,
t0.RATING, t0.REVENUE, t0.STUDIO_ID, t0.TITLE FROM DIRECTOR t1, MOVIE t0 WHERE t1.TALENT_ID
= :talentID   AND t0.MOVIE_ID   = t1.MOVIE_ID   FOR UPDATE" withBindings:{talentID = 87; }>
```

This statement fails with the ORA-01786 error. The Oracle RDBMS doesn't support the use of DISTINCT with the FOR UPDATE clause.

Workaround: If you use the pessimistic locking mode against an Oracle database, you can't use the batch faulting or prefetching features.

Reference: 62425

Problem: Oracle Adaptor doesn't read stored procedures inside of packages.

Description: There is no way to get the database to tell you the components (procedures and functions) that are inside a package definition. Clients can still create stored procedures in the model that will call into packages, it's just that model description using EOModeler won't create these at connect time.

Workaround: You can use EOModeler to create the stored procedure definitions in the model. Just set the external name of the stored procedure to *package-name*.*procedure-name*.

Reference: 73333

Problem: The Oracle adaptor has character set problems on Sparc and m68k.

Description: The Oracle adaptor on Mach does not properly support certain character set conversion on certain architectures:

- On SPARC ISOLatin1 works, but Japanese character sets do not.
- On m68k non-ASCII ISOLatin1 characters are stripped and Japanese character sets don't work at all.

On Intel all character sets work fine, including European and Japanese.

Workaround: To get non-ASCII character sets on SPARC and m68k, install the **EO2JOracleAdaptor.pkg** that's on the Enterprise Objects Framework 2.1 Mach CD.

**Note:** The EO2J Oracle Adaptor is built with the 7.0 version of the Oracle client libraries. Consequently, returning fetch sets from stored procedures is not supported.

Reference: 77990

Problem: Sometimes when you run the **install_database** script or use the schema object creation window in EOModeler, you might get one of the following errors.

```
Mar 27 10:19:34 eoutil[1577] Exception running dump: ORA-00955: name is already used by an existing object
create table   !!! eo_temp_table as select max(MOVIE_ID) counter from MOVIE

Mar 27 10:27:54 eoutil[1591] Exception running dump: ORA-00955: name is already used by an existing object
create procedure   !!! eo_set_sequence is
    xxx number;
    yyy number;
begin
    select max(counter) into xxx from eo_temp_table;
    if xxx is not NULL then
        yyy := 0;
        while (yyy < xxx) loop
            select MOVIE_SEQ.nextval into yyy from dual;
        end loop;
    end if;
end;
```

Description: If the primary key support code dies unexpectedly, it can leave the **eo_set_sequence** stored procedure and the **eo_temp_table** table in the user's schema. This isn't a problem right away, but any subsequent attempt to run the **install_database** script will fail because the statements that attempt to create these schema objects will fail and the data loading will stop before it is finished.

Workaround: Use the following commands from SQLPlus to remove these objects before you attempt to create the schema objects with the **install_database** script.

```
SQL> drop table eo_temp_table;
SQL> drop procedure eo_set_sequence;
```

### Sybase Adaptor

These problems exist in the Sybase adaptor supplied with this release of Enterprise Objects

Framework:

Reference: 62634

Problem: The Sybase adaptor uses **CS_CONVERT()** to convert numeric data into NSDecimalNumbers.

Description: Fetching decimal numbers in locales that use a characters other than `.' for the decimal may not work.

Workaround: None.

Reference: 63169

Problem: Models created for version 4.9 Sybase servers do not include stored procedures.

Description: EOModeler gets stored procedure information using functionality that isn't provided in old servers.

Workaround: Use EOModeler's Stored Procedure Editor to add EOStoredProcedure objects to a model.

Reference: 74633

Problem: You can't link against the Sybase1x framework.

Description: The Sybase1x framework has a bug in its sybase1x.lib, such that anything linked against it attempts to load the nonexistent Sybase.dll (rather than Sybase1x.dll). If you never access an adaptor-specific class such as SybaseChannel, you won't have problem. However, if you do reference an adaptor-specific class, your application (or framework) won't be able to initialize.

Workaround: Make a copy of the Sybase1x.dll named Sybase.dll.

Reference: 75024

Problem: The SQL generation for primary key support in the Sybase adaptor isn't sufficient.

Description: The SQL generated by EOModeler to support primary key generation in Sybase simply invokes the **sp_primarykey** stored procedure. **sp_primarykey** adds useful information to the syskeys table, but doesn't create constraints that enforce uniqueness (or NOT NULL) in the primary key columns.

Workaround: Add the constraints yourself using statements of the following form:

```
alter table <table-name> add constraint primary key (<column-name> {, <column-name })
```

### EOModeler

These problems exist in EOModeler with this release of Enterprise Objects Framework:

Reference: 77880

Problem: EOModeler exhibits odd behavior if you try to paste model objects from a closed window.

Description: Copy a model object (or objects), close the window from which you copied, and then open a new window. The Paste menu item is enabled in the new window, but if you try to paste, nothing happens.

Workaround: Don't close the original window before pasting the model objects.

Reference: 77608

Problem: Existing Enterprise Objects Framework projects don't see the new Enterprise Objects Framework 2.1 source code generation templates (used when you issue the "Generate Source Files..." command in EOModeler).

Description: Enterprise Objects Framework 2.1 includes fixes and additions to EOInterfaceFile.template and EOImplementationFile.template. If a user has a project from Enterprise Objects Framework 2.0 that contains the old versions of these files, the old versions will continue to be used.

Workaround: If you didn't customize the templates, then you should just remove these old files from your project.

If want to build a customized template based on the newest versions, copy /NextDeveloper/Apps/EOModeler.app/Resources/EOImplementationFile.template and /NextDeveloper/Apps/EOModeler.app/Resources/EOInterfaceFile.template into your project directory.

Reference: 77134

Problem: If you open two different Diagram View windows for the same model file, the layout for only one of the windows is saved, and which layout is saved is indeterminate.

Workaround: Only open one Diagram View window for a model at a time.

Reference: 77191

Problem: Pasted attributes do not retain their primary key, used for locking, or class property settings.

Description: If you copy and paste an attribute, the pasted attribute's primary key, used for locking, and class property bits revert to their default settings, regardless of their state in the original. This occurs because the settings are actually maintained by the owning entity, which isn't incorporated into an attribute's pasteboard rendering.

Workaround: Manually set the bits after pasting.

Reference: 63118

Problem: Schema generation may produce identifiers that are too long.

Description: If your database table names are long, the names generated for constraints may be too long.

Workaround: Shorten your table names, or edit the SQL EOModeler generates to shorten the constraint names.

Reference: 51250

Problem: You can't join on derived attributes.

Description: Invalid SQL is generated whenever a derived attribute is designated as a join attribute in a relationship.

Workaround: None.

### On-line Examples

These problems exist in the on-line examples included in this release of Enterprise Objects Framework:

Reference: 65165

Problem: **install_database** doesn't work: Sybase: Can't find type `decimal'.

Description: The **install_database** script in the **EnterpriseObjects/DatabaseSetUp** directory doesn't work for Sybase 4.9 servers.

When the models in the examples directory were converted to Sybase, the server version wasn't taken into account. Some of the attributes were converted to ªdecimalº data type and the Sybase 4.x server doesn't know about these.

Workaround: Use EOModeler to find and change all the external types from decimal to float in each of the models in your examples directory.

Reference: 68231

Problem: **make all** in the Enterprise Objects Framework examples fails sometimes.

Description: The Enterprise Objects Framework examples will fail to make if **${NEXT_ROOT}/LocalDeveloper** is present on your system and you don't have write access to it. The makefile tries to install the BusinessLogic framework in the **/LocalDeveloper/Frameworks** directory.

Workaround: Execute **make** as root or ask your system administrator to create a **/LocalDeveloper/Frameworks** directory and make it writable by you. Alternatively, use ProjectBuilder to change the install directories for BusinessLogic and EOExtensions to a location that's writable by you, then change all the projects that depend on BusinessLogic or EOExtensions to look in the new location.

Reference: 70557

Problem: The Customer.app example doesn't propagate changes to CreditCard objects.

Description: The relationship between CreditCard and Member is not modeled correctly in the examples. CreditCard's primary key is composed of its **cardNumber** and **cardType** attributes. In the Customer application, you can modify the **cardNumber** and **cardType** of a Member's CreditCard, but these changes aren't propagated to the objects to which the CreditCard is related. That is, after saving the changes to the database, the CreditCard's **cardNumber** and **cardType** are updated, but foreign keys in the MEMBER table still have the old values.

CreditCard should have a separate primary key identifier, such as **cardID**, on which the CreditCard-Member relationships are based. This approach would keep CreditCards and Members in sync, and it would also be a more efficient implementation.

Workaround: None.

Reference: 72246

Problem: The ODBC adaptor source code provided as an example is only available on OpenStep for Windows NT.

Description: The source code for the ODBC adaptor is available on Windows NT as an example. It is not, however, included on Mach or PDO.

Workaround: Copy the source code from an NT machine.

**Note:** If you just want the ODBC adaptor to see an example of a concrete adaptor, you can look at the FlatFile adaptor that's included in all releases.

Reference: 76020

Problem: In the Studios.app, buying the movies associated with a Talent doesn't update the Movies' Studios.

Description: The Studio class's **buyAllMoviesStarring:** method doesn't take advantage of Enterprise Objects Framework's EOKeyRelationshipManipulation methods. Instead, it simply invokes [self addToMovies:...].

Workaround: Change **Studio.m**'s **buyAllMoviesStarring:** implementation to the following:

```
- (void)buyAllMoviesStarring:(Talent *)talent
{
    NSEnumerator *movieEnumerator = [[talent moviesStarredIn] objectEnumerator];
    id movie;

    while ( (movie = [movieEnumerator nextObject]))
        [self addObject:movie toBothSidesOfRelationshipWithKey:@"movies"];
}
```

Reference: 69419

Problem: Example data for Movie and Rental models must be installed in different databases for Sybase and Informix.

Description: When installing data for more than one model in a database, the **eo_sequence_table** is only properly initialized for the first model installed.

Workaround: You can either install the data for each model in a different database or change the last line in the **install_database** script located in the DatabaseSetUp directory of your examples directory as follows:

**Change this:**

```
${EOUTIL} dump ../BusinessLogic/Rentals.eomodeld -source plist database
-schemaCreate -postInstall < RentalData.plist
```

**To this:**

```
${EOUTIL} dump ../BusinessLogic/Rentals.eomodeld -source plist database
-schemaCreate -postInstall -force < RentalData.plist
```

When you use the **-force** option as a workaround, exceptions are logged because the **eo_sequence_table** table and **eo_pk_for_table** procedure already exist when you execute **install_databases**. You can just ignore these exceptions.

### Documentation

This problem exists in the documentation included in this release of Enterprise Objects Framework:

Reference: 77170

Problem: Enterprise Objects Framework Developer's Guide sections about ordering database operations are out of date.

Description: The sections "How Do I Order Database Operations" and "Constraints for Enforcing Relational Integrity Rules" in the chapter "Answers to Common Design Questions" in the Enterprise Objects Framework Developer's Guide need to be updated to reflect a new operation ordering algorithm.

The new algorithm works as follows:

- If an entity (Movie) has a to-one relationship to a second entity (Studio) and the inverse relationship is a to-many, then the second entity (Studio) is considered the *master*.
- If an entity (Talent) has a to-one relationship to a second entity (TalentPhoto) and the inverse relationship is also a to-one, then the entities are peers.

Enterprise Objects Framework builds up an entity ordering based on the "master-ness" of the entities. This ordering is built up dynamically and is enhanced as your application touches more and more entities.

Before sending operations to the database, Enterprise Objects Framework orders the operations based on the entity ordering. For inserts, Enterprise Objects Framework orders master entities first; whereas for deletes, Enterprise Objects Framework orders master entities last. Order on updates doesn't generally matter, Enterprise Objects Framework orders masters first.

This new algorithm should reduce the number of scenarios in which you have to reorder operations. However, if your database has sophisticated referential integrity, if it uses triggers, or there are referential integrity constraints that are not modeled as EORelationships, you may still need to reorder adaptor operations programmatically.

## Enterprise Objects Framework Defaults

This section lists the user defaults for which Enterprise Objects Framework checks and changes its behavior.

### Defaults in the NSGlobalDomain

To change these defaults, issue a command such as the following from a shell:

```
defaults write NSGlobalDomain EOAdaptorUseBindVariables NO
```

*EOFDebugEditingContext*

Default is NO.
EOControl Layer will log every time an object is changed.

*EOFDebugUndo*

Default is NO.
Logs each time something is pushed on or popped off an undo list.
Also logs in `[_EOUndoStack dealloc]`.

*EOAdaptorDebugEnabled*

Default is NO.
Logs for connection attempts, all transaction activity (begin, rollback, commit), and SQL statements (select, update, etc.).

*EOProjectSourceSearchPath*

An optional array of paths in which EOAccess searches for framework project directories that contain models.

*EOAdaptorUseBindVariables*

Default is YES.
Controls whether or not the SQL generation process will use bind variables (for Oracle and Informix).

*EOOracleTableNamesSQL*

SQL statement used to identify tables for the model reverse-engineering process.
Default is:

```
SELECT TABLE_NAME FROM USER_TABLES ORDER BY TABLE_NAME")
```

If you would like the model reverse-engineering to look at a different set of the tables, you can provide your own SQL statement, for example:

```
SELECT TABLE_NAME FROM USER_TABLES ORDER BY TABLE_NAME
WHERE TABLE_NAME not in ('MY_USELESS', 'FOO')
```

*EOOracleUseNoWaitLocks*

Default is NO.

Controls whether Oracle uses NOWAIT locks.

*EOSybaseInterfacesFile*

Default is unset.

If the default is set, the SybaseAdaptor initializes the client library to run with this interface the specified interfaces file. This default is not applicable on Windows NT.

*EOSybaseTableNamesSQL*

This default can be used to change the default SQL used to determine the tables that are analyzed in creating a new model.

Default is

```
select name from sysobjects where type = 'U' or type = 'V'
```

*EOSybaseAttributeSQL*

SQL statement used to get the attribute information for a given table.

Default is

```
select a.name attr_name, b.type attr_type, b.name type_name, b.allownulls
allownulls, c.type primary_key, a.length width, a.prec prec, a.scale scale
from syscolumns a, systypes b, syskeys c where a.id = object_id('%@') and
a.id *= c.id and (a.colid *= c.key1 or a.colid *= c.key2 or a.colid *= c.key3
or a.colid *= c.key4) and (c.type = 1 or c.type is null) and a.usertype =
b.usertype and datalength(a.name) > 0 order by a.name
```

*EOSybaseRelationshipSQL*

Default is

```
select a.keycnt, b.name tname, c.name cname, d.name dtname, e.name dcname
from syskeys a, sysobjects b, syscolumns c, sysobjects d, syscolumns e where
(b.type = 'U' or b.type = 'V') and a.depid is not null and a.depid > 0 and
b.id = a.id and b.id = c.id  and ((c.colid = a.key1 and e.colid = a.depkey1)
or (c.colid = a.key2 and e.colid = a.depkey2) or (c.colid = a.key3 and
e.colid = a.depkey3) or (c.colid = a.key4 and e.colid = a.depkey4)) and d.id
= a.depid and d.id = e.id order by b.name, d.name
```

*EOSybaseStoreProcedureSQL*

Default is

```
select name, id from sysobjects where type = 'P'
```

*EOSybaseStoredProcedureDetailsSQL*

Default is

```
sp_sproc_columns %@
```

*InformixINFORMIXSERVER*

No default value.
The name of the Informix server.

*InformixInformixTableNamesSQL*

This default can be used to change the default SQL used to determine the tables that will be analyzed in creating a new model.
Default is

```
select tabname, owner, tabid from informix.systables where (tabid > 99) and
(tabname <> 'ANSI') order by tabname
```

### Defaults in the EOModeler Domain

To change these defaults, issue a command such as the following from a shell:

```
defaults write EOModeler DisableInheritenceCheck YES
```

*BundlesToLoad*

If you write extensions to EOModeler, you can get EOModeler to load them by setting this default. This default is an array, so remember to include parentheses when you set it. For example:

```
defaults write EOModeler BundlesToLoad
"(/LocalDeveloper/ModelerBundles/MyModelerBundle.bundle,
$HOME/eoexamples/ModelerBundle/ModelerBundle.bundle) "
```

*RECORD_FETCH_LIMIT*

Default is 100.
For the Data Browser in EOModeler. This controls how many rows to fetch before asking users if they want to fetch all, fetch another n rows, or just stop now.

*Consistency Checks*

All of the following defaults have the default NO:

DisableConsistencyCheckOnSave
DisableExternalNameCheck
DisableInheritenceCheck
DisablePrimaryKeyCheck
DisableRelationshipCheck
DisableStoredProcedureCheck

*SkipBeautifyNamesOnModelCreation*

Default is NO.
Setting this default to YES will cause EOModeler to leave the entity names and attribute names matching the exact names of their corresponding database objects (including the case).

*DisableAdvancedOptions*

Default is NO.
If set to YES, the EOModeler runs in "Simplified mode". The advanced inspectors don't appear, the choices of table view columns is limited, and so on.

These Release Notes are for informational purposes only. Every effort has been made to ensure the accuracy of information contained in these Notes. However, Apple assumes no responsibility for the accuracy of the information and it is subject to change without notice. Any use of, or actions taken based upon, any information contained in these Notes is done at your own risk.