Release Notes:
Enterprise Objects Framework 2.0

This file contains the most up-to-date release notes for release 2.0 of Enterprise Objects Framework.

The MachOS version of this release is 3-way fat; it can be used to develop software for NeXT, Intel, and SPARC. The OpenStep for Windows version of this release can be used to develop software for Intel machines running Windows NT.

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**.

In order to use Enterprise Objects Framework on Windows NT, you must purchase appropriate client libraries from your database vendor.

Here's what you need:

Oracle
Phone: (800) 542-1170
Ask for SQLNET V2.2 for PC/Windows NT

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

Sybase
Phone: (800) 685-8225
Ask for: Open Client Runtime for PC/Windows NT

Informix
Phone: (800) (800)331-1763
Ask for: ESQL/C for Win32

Converting Release 1.x Applications to 2.0 (Mach only)

Enterprise Objects Framework 2.0 for Mach includes **tops** scripts for performing source code conversion of your Enterprise Objects Framework 1.x applications to 2.0. Because of the magnitude of changes in the architecture of the 2.0 release, these scripts are by no means sufficient for automatically converting your application. However, once you've decided how the structure of your application will evolve to take advantage of the new architecture, these scripts can be useful in identifying and sometimes making specific API conversions. The scripts, along with instructions for running them, can be found in **/NextDeveloper/EOF1xTo2Conversion**.

For more information on the differences between Enterprise Objects Framework 1.x and Enterprise Objects Framework 2.0, see the document *Differences Between Enterprise Objects Framework 1x and 2.0.* You can find it on-line in

/NextLibrary/Documentation/EnterpriseObjects/1x_To_2.

Installing the Examples

This release provides on-line examples to help familiarize you with Enterprise Objects Framework 2.0. These examples are located in **/NextDeveloper/Examples/EnterpriseObjects**. Installing the examples involves these steps:

- Setting up users and databases on your database server for the example databases.
- Installing the example directory.
- Populating your database server with example data.

Setting up Database Accounts

The Enterprise Objects Framework 2.0 examples use two sets of tables: Movies and Rentals. Some examples use just one of the these databases, while others use both. The multi-database support in Enterprise Objects Framework 2.0 makes it possible for you to install these databases in three different configurations:

- Both sets of tables together in a single user/database.
- Each set of tables in its own user/database on the same database server.
- Each set of tables on its own database server (for example, Movies on Informix, Rentals on Oracle).

Depending on your desired setup, you use the tools available with your database server to set up one or two new user/databases. For example. on Sybase you might create a new database on your server called ªMoviesº and login with the user ªsaº. On Oracle you might create a new user with the name ªMoviesº. Once you have set up these accounts, you're ready to install the examples.

Copying the Example Directory

To configure and build the examples you need to copy the example directory to a writable area in your file system. You can do this by copying the **/NextDeveloper/Examples/EnterpriseObjects** folder into your home directory (or any other directory writable by you).

Configuring the Example Models

The model files used by the examples must be configured to use your adaptor and server. To configure the examples, run the **configure_examples** program in your copy of the examples directory. It will ask you for the name of the adaptor you wish to use (Informix, Oracle, Sybase, and so on) and for the login information for your database. It will then convert the example models for your server.

Populating the Databases

Now that the examples are configured, you can fill your example databases with sample data. The **install_database** tool in the **DatabaseSetUp** directory will connect to your databases, add the example tables, and fill them with data. If you later wish to remove the data, simple run the **drop_database** tool.

Building the Example Programs

With your example projects installed and your database filled with data, you are ready to build and run the examples. To do this, in a command shell **cd** to your example directory and type **make all**. This performs a **make install** on **BusinessLogic.framework** and **EOExtensions.framework** to put them in **/LocalDeveloper/Frameworks**, where they are shared by many of the other examples. It then makes all of the examples applications.

Note: The **/LocalDeveloper/Frameworks** directory must be created and writable by you in order to build the examples. On Mach this can be accomplished by using **su** to become the superuser and then executing the following commands:

```
mkdirs /LocalDeveloper/Frameworks
chmod a+w /LocalDeveloper/Frameworks
```

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 attempst, all transaction activity (begin, rollback, commit), and SQL statements (SELECT, UPDATE, etc.).

EOProjectSourceSearchPath

An optional array of paths in which EOAccess will search 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 (default is "SELECT TABLE_NAME FROM USER_TABLES ORDER BY TABLE_NAME")

If you would like the model re-engineering process to look at a subset of the tables in the given schema, you can provide your own SQL statement like "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 this default is set, the SybaseAdaptor will initialize the client library to run with this interface 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 will be 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
"(/NextDeveloper/Apps/EOModeler.app/Resources/DiagramEditor.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 these 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 EOModler runs in "Simplified mode". The advanced inspectors don't appear, the choices of table view columns is limited, and so on.

New Features for 2.0

- A new class, EOEditingContext, that manages an object graph and coordinates change notification. This class is also responsible for change tracking and undo. One EOEditingContext can also be nested inside another, thereby supporting nested in-memory transactions.
- Support for multiple databases.
- NSDecimalNumber, a class that provides a lossless representation of decimal numbers.
- User interface formatting and validation.
- Updatable relationships: Your business logic may now update relationship pointers in the object graph and the framework will automatically generate the appropriate foreign keys changes to affect the change in the database.
- Referential integrity rules. You can explicitly add referential integrity rules in EOModeler. They will be enforced by Enterprise Objects Framework in your running application.
- Improved support for modeling inheritance.
- Enterprise object±based validation.
- Update support for flattened attributes.
- Schema creation. You can use the Generate SQL command in EOModeler to generate SQL for a selected entity or entities.
- Automatic primary key generation: The Framework can now automatically assign primary keys for newly inserted objects.
- Structured qualifiersÐthat is, qualifiers are now independent of database and SQL.
- Incremental model loading for improved performance. Models now consist of an **.eomodeld** directory in which each entity has its own file. Entity descriptions are

loaded into your application only as needed.

- Support for Informix servers.
- An ODBC adaptor (Windows NT only).
- Improved support for Oracle stored procedures.
- Reverse engineering of primary keys and relationships on all supported databases (Informix, Oracle, and Sybase). Enterprise Objects Framework can now read this information from the database and incorporate it into a model.
- The Sybase adaptor is now based on CT-Lib.
- The ability to edit the UserInfo dictionary in EOModeler.

For more discussion of these features, see the document *Differences Between Enterprise Objects Framework 1x and 2.0.*

New Features Shared with Enterprise Objects Framework 1.2

- An Informix adaptor is included in this release.
- Adaptors are now frameworks. Just as with the old bundle-based adaptors, framework adaptors can be dynamically or statically linked. If you need to reference a class defined in a particular adaptor from your application code (for example, the InformixChannel class), you have to add the adaptor to the Frameworks list for your project.

  If you have custom adaptors that you're converting to 2.0, you need to convert them into frameworks. If you do this, note that you may encounter problems linking to the client libraries that weren't present when you were loading your adaptor as a bundle. For example, you can't use common symbols. If you use undefined symbols, you need to set a flag in the linker to tell it to ignore those symbols.

- EOAdaptor's use of **adaptor.info** has been removed. Instead, the **Info-*platform*.plist file** from the adaptor's bundle is consulted for the adaptor class name and adaptor major version number. The key names are as follows:

  EOAdaptorMajorVersion
  EOAdaptorClassName

  To add key-value pairs to the infoDictionary, add an ASCII property list file named **CustomInfo.plist** to the supporting files category of your project. The project makefiles will automatically merge this into the **Info.plist** that is derived from your project.
- The following Enterprise Objects Framework 1.1 classes have been renamed and moved into the AppKit: NXTableView (renamed to NSTableView), and NXImageView (renamed to NSImageView).
- EOApplication is obsolete. NSApplication now manages the autorelease pool.

Login Panel Loading Changes

The Enterprise Objects Framework version 1.2 and 2.0 handle login panels differently than version 1.1 in order to factor them into separate bundles, thereby eliminating any link dependency on the AppKit in both the adaptors themselves and in EOAccess. This change allows a single adaptor binary to be used in both OpenStep for Windows and D'OLE. To convert your Enterprise Objects Framework 1.1 adaptor to the new API you must:

- Create a new bundle subproject called "LoginPanel" in your adaptor project, and move the login panel source files and nib file there.

- Remove the AppKit from the list of frameworks for the main bundle and add it (along with EOAccess and Foundation) to the **LoginPanel.bproj**.

- Change your LoginPanel class to subclass from EOLoginPanel rather than NSPanel.

- Change your login panel nib file so that the login window is a subclass of NSPanel (not your EOLoginPanel subclass). Change your file's owner to be the EOLoginPanel subclass.

- Implement `- (NSDictionary *)runPanelForAdaptor:(EOAdaptor *)adaptor validate:(BOOL)validate`

  See **/NextDeveloper/Examples/EnterpriseObjects/FlatFileAdaptor/LoginPanel.bproj/FlatFileLoginPanel.m** for an example implementation).

New Methods and Behavior

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

Locating the Documentation for this Release

The documentation for this release is provided in RTF (readable on Mach only) and in PDF format to be readable (and printable) on all platforms. On Windows NT, documentation is available in the ªOpenStep Books Onlineº item of the OPENSTEP Enterprise program group. You can also find the documentation in the following on-line locations:

- The *Differences Between Enterprise Objects Framework 1x and 2.0* document describes how the product has changed between the 1x and 2.0 releases. See **/NextLibrary/Documentation/EnterpriseObjects/1x_To_2**.
- The *Enterprise Objects Framework Developer's Guide* provides a conceptual overview of the product. In addition to the hard copy book, a pre-production version of the book is provided on-line in PDF format. See **/NextLibrary/Documentation/NextDev/EnterpriseObjects/Guide**.
- The *Enterprise Objects Framework Reference* includes class specifications for the Enterprise Objects Framework classes. Reference documentation is located in **/NextLibrary/Frameworks** in one of these locations:

  /EOAccess.framework/Resources/English.lproj/Documentation/Reference
  /EOInterface.framework/Resources/English.lproj/Documentation/Reference
  /EOControl.framework/Resources/English.lproj/Documentation/Reference

If You Need a PDF Reader...

If you need a PDF Reader, you can download one from one of the following web locations:

- For Windows or Solaris: **http://www.adobe.com/acrobat/**
- For NEXTSTEP: **http://www.BITart.com/PDFView/**

Other Documentation

There are other documents that may provide useful information to you, depending upon your application:

- The *OpenStep Conversion Guide* discusses how to use the automated conversion process to convert applications to OpenStep. See **/NextLibrary/Documentation/NextDev/Conversion/ConversionGuide**.

Documentation Feedback

Your comments on our documentation are especially valuable. Please send electronic mail with your comments and suggestions to **techpubs_feedback@next.com**. Please include **EODoc:** as the first part of your message's subject.

Known Problems in Release 2.0

Enterprise Objects Framework

These problems exist in this release of Enterprise Objects Framework:

Reference: 73506

Problem: Adding categories to the concrete adaptor classes (such as SybaseAdaptor) breaks the login panel on NT.

Description: If you add a category to a concrete adaptor class, whenever your application encounters an invalid connection dictionary the login panel won't appear.

Workaround: If you add a category to a concrete adaptor class, subsequently invoking **bundleForClass:** returns the bundle in which your category lives. Unfortunately, Enterprise Objects Framework uses this method to load the LoginPanel bundle. To work around this problem you need to copy the **LoginPanel.bundle** from the adaptor resources directory (such as **SybaseEOAdaptor.framework/Resources**) into the resources directory of the application or framework to which you added the category.

Reference: 73506

Problem: EOActionAssociation doesn't present an error when it's given a bad selector.

Description: If the user types the wrong selector for the action key, the action association simply does nothing when the button is pressed. It should instead raise an exception or log so that developers have a better chance of detecting the typo.

Workaround: When a button wired with an EOActionAssociation doesn't trigger the expected behavior in an application, double-check that the key exactly matches the selector name.

Reference: 73448

Problem: When you send an EOEditingContext an **insertObject:** message immediately followed by a **deleteObject:** message, the EOEditingContext does not forget the object.

Description: Deleting newly inserted objects from an EOEditingContext does not remove them from the registered objects list. This is a problem because they can then be edited (for example, a relationship they are set on can be reset, causing the reverse, outbound relationship to be updated). Editing these objects puts them in the updated objects list, but they don't have a global ID, so the EOEditingContext raises an exception when it is sent **saveChanges** (this is good, because the deleted objects shouldn't be in the updated list in the first place).

Workaround: Send the EOEditingContext a **processRecentChanges** message before invoking **deleteObject:**.

Reference: 73414

Problem: The EODisplayGroup delegate method **displayGroupDidChangeSelection:** isn't invoked if a different object is selected under the same selection indexes.

Description: The EODisplayGroup delegate method, **displayGroupDidChangeSelection:** is invoked only when the list of selected *indexes* changes, not when the list of selected *objects* changes. So if the selection is at index 0 and a new object is inserted at index 0 and selected, the delegate is not called. This can be a problem in an inheritance application like Inventory, which displays a different sub-panel depending on the class of the selected object. When inserting a new object, the selection index may stay the same but a different object (the new one) is selected. If the new object is of a different class than the old, an exception will likely result when the old (inappropriate) sub panel refreshes.

Workaround: The delegate can register as an observer of the EODisplayGroup so that it gets **subjectChanged** messages when the contents have changed. It can then check that the **selectedObjects** array has changed, and if so forward itself the **displayGroupDidChangeSelection:** message.

Reference: 73219

Problem: Setting a delegate on an EOEditingContext that doesn't implement the method **editingContext:shouldPresentException:** causes exceptions to not be presented.

Description: If an EOEditingContext is assigned a delegate but that delegate doesn't implement **editingContext:shouldPresentException:**, then the EOEditingContext acts as if the method was implemented to return NO. This is incorrect; it should have the same behavior as when the delegate is not set: exceptions should be presented (via the message handler).

Workaround: When setting an EOEditingContext delegate, always implement **editingContext:shouldPresentException:** to return YES.

Reference: 72933

Problem: When the user framework adds a category to Enterprise Objects Framework classes, Enterprise Objects Framework fails to find **License.table** file.

Description: Enterprise Objects Framework verifies that certain features are licensed by looking up a **License.table** file in the the Resources directory of the EOAccess bundle. If a user application includes a framework that adds categories on certain Enterprise Objects Framework classes, Enterprise Objects Framework will incorrectly look for the license file in the user framework rather than EOAccess, resulting in Enterprise Objects Framework considering the feature as unlicensed

Workaround: Copy the **License.table** file from **/NextLibrary/Frameworks/EOAccess.framework/Resources** into the resources of the framework project that adds the category to Enterprise Objects Framework. *** Note that it is in violation of your software license agreement to use this technique to enable features of Enterprise Objects Framework for which you are not legally licensed.

Reference: 72903

Problem: Undo/Redo doesn't record unsuccessful deletions correctly

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 will need 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: 72795

Problem: The "New Editor..." menu item is missing on NT .

Description: The "New Editor" menu item is not present in the NT nib file. "New Editor" allows you to have two different windows open on the same document. This is helpful when you need to compare two entities.

Workaround: Since the implementation exists in EOModeler, you can actually just edit EOModeler's main nib on NT and get this working:

1. Go to $(NEXT_ROOT)/NextDeveloper/Apps/EOModeler.app/Resources/English.lproj
2. Make a backup copy of EOModelerNT.nib. In a shell type:

```
cp -r EOModelerNT.nib EOModelerNT.nib.original
```

3. Use Interface Builder to open the nib.
4. Use the "Classes" folder tab to add the **newEditor:** action to the EOModeler class.
5. Add a "New Editor" menu item in the "Tools" menu. Put it just above the "Preferences..." item.
6. Create the target/action connection from this new menu item to the **newEditor:** action in "File's Owner".

Reference: 72714

Problem: In-memory descending sort on a property than contains **nil**s does not work correctly.

Description: When you use EOSortOrdering to perform a descending in-memory sort of an array of objects in which the sorted property may contain **nil**, objects containing **nil** are sorted inconsistently.

Workaround: Add the following category to your application:

```
@implementation NSObject (_FIXEOSortOrderingComparison)
- (NSComparisonResult)compareDescending:(id)other
{
    NSComparisonResult result;

    if (other == [EONull null]) return NSOrderedAscending;
    result = [(NSString *)self compare:other];
    return (result == NSOrderedSame) ? result : -result;
}
@end
```

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. However, if an EOAdaptor has been created and assigned a delegate, newly-created EOAdaptorContexts and EOAdaptorChannels are not automatically assigned the delegate. However, if a new delegate is assigned to an EOAdaptor, it's propagated to any of that adaptor's existing contexts and channels.

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: 72603

Problem: Column appears twice in SQL insert statement.

Description: This happens whenever a sub-entity has a flattened to-one relationship that is also in the parent entity. In this case, the column of the source attribute of the relationship appears twice in the insert statement.

Workaround: Ensure that the source attribute in the flattened to-one relationship in the sub-entity is *not* marked as a primary key, usedForLocking, or as a class property. This workaround has the adverse effect that the attribute will not be used for locking purposes.

Reference: 72531

Problem: EOEditingContext leaks object record structs when forgetting objects.

Description: Whenever an object is forgotten (unregistered) from an EOEditingContext (for example, when releasing the EOEditingContext), an internal structure is leaked.

Workaround: This bug is patched in WebObjects 3.0. OpenStep applications can plug this leak by executing the following statements during application initialization:

```
[EOEditingContext _enableFreeingObjectRecords:YES];
[EOUndoManager _setFreesUndoStacks:YES];
```

Reference: 72527

Problem: Foreign keys are not inserted into database as expected.

Description: In the case where an entity contains an attribute that is not marked as usedForLocking, as a class property, or as a primary key, and the attribute isn't used in any relationship on that entity, then the attribute won't be included in the SQL insert statement.

Workaround: Mark the attribute as used for locking.

Reference: 72483

Problem: To-many relationship properties in newly inserted objects aren't being properly initialized.

Description: Newly inserted objects should automatically get empty arrays assigned to their to-many relationship properties, so that when these properties are subsequently sent an **addObject:** message, everything works correctly.

Workaround: In the **awakeFromInsertionInEditingContext:** method of your enterprise object class, initialize variables that represent to-many relationships with new NSMutableArrays.

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: 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: 72027

Problem: PDO: EOKeyValueCoding is broken for doubles on HPUX

Description: EOKeyValueCoding is not working 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 to use NSNumber *, NSDecimalNumber *, or **int** accessors.

Reference: 71553

Problem: EOPickTextAssociation displays an alert panel whenever the '[' character is entered.

Description: EOPickTextAssociation should either escape all substitution characters or wait until a complete expression is typed in before executing. Since it does not, EOQualifier thinks that it has found an illegal pattern and raises an exception, which the EOPickTextAssociation displays in an alert panel.

Workaround: None.

Reference: 70937

Problem: Validation methods don't check relationship optionality.

Description: Relationships can be marked as optional or mandatory in the Relationship Inspector of EOModeler. The validation methods supplied by Enterprise

Objects Framework should verify that all relationships that are marked as mandatory have at least one object in them, but this property is ignored.

Workaround: Implement the check in your own validation method.

Reference: 70847

Problem: After disconnecting all EOColumnAssociations from a table view, you can't connect to a different EODisplayGroup.

Description: If the columns are connected via EOColumnAssociations to one EODisplayGroup, disconnected, and then you attempt to connect a column to a different EODisplayGroup, all of the EODisplayGroup's properties appear as gray in the Inspector-- a connection is not allowed.

Workaround: This is actually a feature! When the first EOColumnAssociation was created, an EOTableViewAssociation was bound to the table itself. This association remains connected to the original EODisplayGroup even after the column has been disconnected, and this connection prevents columns from being connected to a different EODisplayGroup. To make the new connection, click on the table view and disconnect the EOTableAssociation.

Reference: 70260

Problem: Some validation error messages returned by the Framework have not been localized.

Description: Some validation 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) are not localized.

Workaround: Implement an error handler for use with the EOEditingContext and have that handler look for the localized string in a file that the programmer has created.

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: Verify that no two models have the same entity name in them either manually or by writing a method to perform the check.

Reference: 69934

Problem: EOQualifier needs a case insensitive 'like' operator.

Description: EOQualifier needs a case insensitive 'like' operator that works both in memory and in SQL.

Workaround: You can use a pattern like '*[Hh][Ee][Ll][Ll][Oo]*' or you can create a method on the enterprise object that always returns a version in uppercase and use that method as a key in the qualifier.

Reference: 69678

Problem: Enterprise Objects Framework does not insert or delete rows in the database in a way that is compatible with the database's referential integrity constraints.

Description: Enterprise Objects Framework performs its database operations in an order that causes the database's referential integrity constraints to be violated.

Workaround: Implement the **databaseContext:willOrderAdaptorOperationsFromDatabaseOperations**

: method on the EODatabaseContext and order the adaptor operations so that they are compatible with the database's referential integrity constraints. For more discussion of this subject, see the section "How Do I Order Database Operations?" in the chapter "Answers to Common Design Questions" in the *Enterprise Objects Framework Developer's Guide*.

Reference: 69670

Problem: When you use the 'like' operator in a format string that's specified as an argument to the EOQualifier method **qualifierWithQualifierFormat:**, the operator is only recognized when it is all lowercase

Description: This method raises an exception when given a format string such as: "name LIKE 'fred'" because it only recognizes the LIKE operator when it is in lower case letters.

Workaround: Use 'like' instead of 'LIKE'.

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: 46679

Problem: All private instance variables should be **@private** or begin with _.

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

Workaround: Don't access Enterprise Objects Framework instance variables in client programs.

Reference: 46967

Problem: Using delete propagation triggers in database can cause Enterprise Objects Framework to become out of sync.

Description: The use of delete propagation triggers in the database instead of the model may cause Enterprise Objects Framework to become out of sync. This becomes a bigger problem if the Framework is executing several operations as part of a transaction and the transaction is rolled back by a trigger rule.

Workaround: See the chapter ªAnswers to Common Design Questions,º in the *Enterprise Objects Framework Developer's Guide* for information on handling this problem.

Reference: 47202

Problem: Conversion from money to double doesn't always work.

Description: Database number types are sometimes more precise than a C **float** or **double**. Consequently when these numbers are read in from the database, stored in a double, and then written back out to the database, rounding errors may appear. For instance, the value `2078927178.0000' is converted as `2078927178.0000002384185791'.

Workaround: Use the NSDecimalNumber class for money.

Reference: 47832

Problem: Don't mark as ªused for lockingº those attributes in the model with a custom data type unless the custom value class implements a value-based **isEqual:** method.

Workaround: Implement **isEqual:** on the custom data type and don't mark it used for locking.

Reference: 51250

Problem: You can't join on derived attributes.

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

Workaround: None.

Reference: 52557

Problem: Enterprise Objects Framework wrongly generates **t0.*nil***.

Description: Enterprise Objects Framework sometimes inappropriately generates SELECT statements that include text like ªt0.*nil*º. For example, this can happen when you make a column with no column name, a column with a column name like ªconvert(varchar(12), 'Default')º, or derived attributes with no column name.

Workaround: None.

Reference: 53598

Problem: Enterprise Objects Framework doesn't support to-one relationships joining on candidate (partial primary) keys in the destination.

Description: Enterprise Objects Framework doesn't support to-one relationships that join on only part of a compound primary key of the destination entity. This is because the Framework bases its uniquing of fetched objects on their full primary keys. If Enterprise Objects Framework fetches an object with a to-one relationship, it must look that object up in the uniquing table to see if it is already in memory. If it isn't, Enterprise Objects Framework must create a fault for it. Because objects are uniqued by their full primary keys, a candidate key is insufficient to do the lookup. Even doing lookups by both full primary key and by candidate key doesn't work, because a fault created for one relationship joined on the full primary key can't be correlated with a fault for another relationship joined to the same row based on the candidate key.

Workaround: Use a to-many relationship and define accessor methods for the enterprise object that emulate the to-one relationship by getting and setting the first object of the relationship array.

Reference: 61475

Problem: When performing a inheritance fetch, we need to group the fetches by tables.

Description: Enterprise Objects Framework does not currently take advantage of the union table method of inheritance mapping to perform a deep fetch of an inheritance hierarchy via a single fetch of all columns. Intead, it currently executes multiple fetches against the same table.

Workaround: The entire hierarchy can be mapped as a single entity and then EOModelGroup delegate methods can be used to tell Enterprise Objects Framework how to allocate the proper subclasses.

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 and then insert a few objects and delete a few objects, and update a few objects, telling the EOEditingContext to revert will back out the updates, but not insertions or deletions.

Workaround: 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 insertion or deletions). The easiest workaround is to programatically tell all affected EODisplayGroups to refetch after telling the EOEditingContext to revert.

Reference: 64283

Problem: Enterprise Objects Framework 2.0 examples merge with obsolete Enterprise Objects Framework 1.1 examples.

Description: If a system has Enterprise Objects Framework 1.1 Developer installed on it (EODeveloper.pkg) and then upgrades to 4.0, the EODeveloper package (incorrectly) stays installed, and the examples stay in **/NextDeveloper/Examples/EnterpriseObjects**.

When the user later installs EO2Developer, the Enterprise Objects Framework 2.0 examples are merged with the Enterprise Objects Framework 1.1 example files already in **/NextDeveloper/Examples/EnterpriseObjects**.

Workaround: Uninstall 1.1 EODeveloper.pkg before installing EO2Developer.pkg.

Reference: 64968

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 whenever an external type has been entered erroneously that does not actually map to a column type recognized by the database. It may also occur when a Sybase model has no record of user defined types that have been used in the model.

Workaround: In the case of an erroneous external type definition, simply correct it. In the case of a user defined type, create a new model by reverse engineering the database, and copy the connection dictionary with the user defined type information from the new model to the original one.

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: 68146

Problem: Fault failure leaves EOInterfaceLayer unstable.

Description: If an entity being displayed has a to-one relationship and the row with the corresponding foreign key cannot be found in the database, when you try to access the source object your user interface gets into a corrupted state.

Workaround: Enterprise Objects Framework raises an exception when a to-one relationship cannot be resolved (because of a referential integrity problem in the database). If this exception is raised by user interface code (for example, in displaying a master-detail relationship) the state of the user interface control objects can get corrupted. To prevent this you must prevent the exception. One way is to trap the exception by implementing the EODatabaseContext delegate method **databaseContext:failedToFetchObject:globalID:**. The other is to use a to-many relationship in your model instead of a to-one.

Reference: 68604

Problem: Repeated false alert that Enterprise Objects Framework 1x nib file is being converted to Enterprise Objects Framework 2.0.

Description: When you reopen a nib file after conversion you get the message: ªThis Enterprise Objects Framework 1x nib file has been translated to Enterprise Objects Framework 2.0. If you did not intend to convert to Enterprise Objects Framework 2.0, unload EOPalette, quit Interface Builder without saving this file, and reopen the nib.º

Workaround: Quit Interface Builder, then reopen the nib file.

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 "EOF Application," this code is automatically added to your project main file.

Oracle Adaptor

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

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: No workaround for this problem on EOF 2.0 is currently avaialable.

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: 63348

Problem: Converted models have no width informationÐOracle fetches empty strings.

Description: The Enterprise Objects Framework 2.0 Oracle Adaptor has problems if you attempt to use an eomodel file from the Enterprise Objects Framework 1.0 release. These model files don't contain width information for string values, which results in all the strings being fetched with 0 length. If this happens for the column that is the primary key, you will also have problems with uniquing since every row will appear to have the same key.

Workaround: Use EOModeler to add width information for VARCHAR2 and RAW columns.

Sybase Adaptor

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

Reference: 74943

Problem: Unable to correctly reverse-engineer schemas whose tables have more than 4 primary key components.

Workaround: Use the following defaults command executed in a shell-- note the use of key5 - key 8; the reason it stops at 8 is that Sybase limits the number of keys to 8:

```
defaults write NSGlobalDomain EOSybaseAttributeSQL '"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.colid and (a.colid *= c.key1 or a.colid *= c.key2 or a.colid *= c.key3 or a.colid *= c.key4 or a.colid *= c.key5 or a.colid *= c.key6 or a.colid *= c.key7 or a.colid *= c.key8) and (c.type = 1 or c.type is null) and a.usertype = b.usertype and datalength(a.name) > 0 order by a.name"'
```

Reference: 74633

Problem: Can't link against the Sybase1x framework.

Description: The Sybase1x framework has a bug in its sybase1x.lib, such that anything linked against it will incorrectly attempt to load the nonexistent Sybase.dll (rather than Sybase1x.dll). If you just write code to the EOAdaptor generic level, things will work fine, but if you reference anything like SybaseChannel or SybaseColumn you won't be able to initialize your application (or framework).

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

Reference: 73321

Problem: You can't update a BLOB and non-BLOB value in the same row.

Description: If you attempt to update a BLOB attribute and a non-BLOB attribute in the same row, the update to the non-BLOB attribute will succeed, but the update to the BLOB will fail because the adaptor attempts to get the io_desc structure with a SELECT statement that includes a WHERE clause for the old (pre-update) non-BLOB value. Thus, the adaptor doesn't get the io_desc, and the transaction rolls back.

Here is the SQL from an example in which the attributes "name", "image1", and "image2" have been changed:

```
Sep 27 14:21:43 trey[3138]   *** [<SybaseChannel: 0xb3dbc> executing
SQL: UPDATE   TREY_TEST   SET   NAME = 'test', IMAGE2 = 0xFFFF,
IMAGE1 = 0xFFFF   WHERE   (PK = 24   AND   NAME = 'foo')]
Sep 27 14:21:43 trey[3138]   *** [<SybaseChannel: 0xb3dbc> executing
SQL: SELECT   t0.IMAGE1 FROM TREY_TEST t0 WHERE (t0.PK = 24
AND   t0.NAME = 'foo')]
```

Clearly, the name should no longer be 'foo' since it was just updated it to 'test'.

Workaround: Change the database schema to put the BLOB's into their own table accompanied only by a primary key that the adaptor doesn't update.

Reference: 73136

Problem: If an entity has more than 1 BLOB, updating one of the values will cause the other(s) to be set to 0xFFFF.

Description: When the Sybase adaptor is updating a row in which a BLOB value has changed, it first sets the value of the BLOB to 0xFFFF, and then it SELECT's the row in order to get the io_desc structure so it can write the BLOB. Unfortunately, the code that replaces the updated BLOB values with 0xFFFF is also replacing the BLOB values that didn't change with 0xFFFF.

Workaround: In order to work around this problem, you need to implement one of the delegate methods that gets called during the generation of SQL for updating the rows. See NeXTanswer #2475 for an example.

Reference: 72641

Problem: The Sybase DLL is not found when you run Enterprise Objects Framework 2.0 applications on Windows.

Description: The Sybase DLL is not furnished with the Enterprise Objects Framework 2.0 installation packages.

Workaround: The Sybase DLL and the necessary associated files are available through NeXTanswer #2471 (a README) and NeXTanswer #2472. You must download and install these files.

Reference: 61481

Problem: When debugging, the new Sybase CT-Lib adaptor blocks if you you try to step over a call that requires a trip to the Sybase server.

Description: GDB hangs when you use the **next** command to step over a line of code which eventually causes a call to the Sybase client library.

Workaround: Set a breakpoint after the line and use **continue** rather than **next**.

Reference: 61999

Problem: An update fails for text that disallows NULL and has an empty field.

Description: Whenever a text or image column is being updated, the Sybase adaptor attempts to ensure that the column has been initialized correctly by assigning NULL to it. In cases where null is not allowed this does not work.

Workaround: Modify the table definition to allow nulls.

Reference: 62634

Problem: The Sybase adaptor is using **CS_CONVERT()** to convert numeric data into NSDecimalNumber.

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

Workaround: None.

Informix Adaptor

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

Reference: 74201

Problem: The Informix adaptor doesn't support the Japanese version of Informix.

Description: The Informix adaptor is based on a version of the Informix client library that doesn't include internationalization support.

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.

Reference: 63225

Problem: Informix adaptor does nothing for **returnValuesForLastStoredProcedureInvocation**.

Description: The Informix adaptor will not return variable values for stored procedures.

Workaround: Don't use stored procedures. If you need to access their logic, use an intermediate table to hold return values and then SELECT them back to your client.

Reference: 64031

Problem: UserDefaults are different between Enterprise Objects Framework1.2 and Enterprise Objects Framework2.0 InformixAdaptors.

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

Enterprise Objects Framework 2.0 stores defaults in the standard NSGlobalDomain and prefixed all keys with ªInformixº, that is: InformixINFORMIXDIR, InformixDBDATE, InformixDBLANG, InformixDBMONEY, InformixLogErrors, InformixShowSystables, InformixShowTableOwner, InformixBeautify,, InformixDefaultIsolationLevel, InformixGlobalOptimization, InformixGlobalExplain, InformixGlobalLockMode, InformixGlobalPDQPriority, InformixGlobalDataSkip, InformixGlobalConstraints, InformixDatabaseExclusive.

Workaround: None.

ODBC Adaptor

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

Reference: 73072

Problem: ODBC cannot save custom classes

Description: When you try to save a custom class (such as NSImage), an exception is raised with the message "NSImage does not respond to selector length raise." ODBC does not appear to be converting these classes to NSData like the other adaptors.

Workaround: None. A fixed version of the ODBC adaptor will be made available through NeXTanswer #2477.

For anyone who is implementing an adaptor based on the ODBC example, here is the corrected source:

```
@implementation ODBCByteColumn
...
- (void)takeInputValue:(id)value
{
    // Check for nulls
    if(value == [EONull null]) {
        _returnedLength = SQL_NULL_DATA;
        return;
    } else {
        id primValue;
        const void *bytes;
        int length;
        primValue = [_attribute adaptorValueByConvertingAttributeValue:value];
        if ([primValue isKindOfClass:[NSString class]])
            primValue = [primValue dataUsingEncoding:_encoding allowLossyConversion:YES];

        if (![primValue isKindOfClass:[NSData class]])
            [NSException raise:EOGeneralAdaptorException format:@"unable to create a byte representation for class
                    \"%@\"", [primValue class]];

        bytes = [primValue bytes];
        length = [primValue length];
        memcpy(_value, bytes, length);
        _returnedLength = length;
    }
}

@end

@implementation ODBCLongByteColumn
...
- (void)takeInputValue:(id)value
{
    // Check for nulls
    if(value == [EONull null]) {
        _returnedLength = SQL_NULL_DATA;
        return;
    } else {
        id primValue;
        const void *bytes;
        int length;
```

```
            primValue = [_attribute adaptorValueByConvertingAttributeValue:value];
            if ([primValue isKindOfClass:[NSString class]])
                primValue = [primValue dataUsingEncoding:_encoding allowLossyConversion:YES];

            if (![primValue isKindOfClass:[NSData class]])
                [NSException raise:EOGeneralAdaptorException format:@"unable to create a byte representation for class
                        \"%@\"", [primValue class]];

            bytes = [primValue bytes];
            length = [primValue length];

            if (_valueLength < length) {
                [self freeValue];
                _valueLength = length;
                [self allocateValue];
            }
            memcpy(_value, bytes, length);
            _returnedLength = length;
        }
    }

@end
```

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 or use the FlatFile adaptor (included on all releases) instead.

Reference: 70193

Problem: The ODBC adaptor does not support the stored procedure API.

Workaround: Subclass the adaptor and implement the functionality.

EOModeler

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

Reference: 73684

Problem: When you convert a model from 1.x to 2.0, any relationship information in the user dictionary is lost.

Workaround: Before opening the 1.x model in EOModeler 2.0, open a copy of it in Edit and perform a Find and Replace to convert "userDictionary" to "userInfo". Open the modified copy of the model in EOModeler.

Reference: 73585

Problem: In EOModeler on Windows NT, the "Preferences..." menu item is always disabled.

Workaround: One solution is to use Interface Builder to open the EOModeler NT nib file and make a target connection from the "Preferences..." menu item to File's Owner **preferences:** action.

Alternatively, you can use use the defaults database to control everthing in the Preferences panel, by issuing a command such as the following in a shell:

```
default write EOModeler DisableStoredProcedureCheck YES
```

You can use this technique to change all of the settings (listed below) that are normally modified through the Preferences panel. Note that the default setting for all of these items is NO.

RecordFetchLimit (number)	Limits the number of rows fetch before the Data Browser asks you if you want to fetch more.
DisableAdvancedOptions (BOOL)	Run EOModeler in simplified mode
DisableConsistencyCheckOnSave(BOOL)	Don't check consistency before saving
DisablePrimaryKeyCheck(BOOL)	Don't check for primary keys
DisableRelationshipCheck(BOOL)	Don't check relationship definitions before saving
DisableExternalNameCheck(BOOL)	Don't check for external names before saving
DisableInheritenceCheck(BOOL)	Don't check for inheritance problems
DisableStoredProcedureCheck(BOOL)	Don't check for valid stored procedure usage

Reference: 73064

Problem: Putting an invalid name in the Parent column raises an exception.

Description: In the table mode view of entities in EOModeler, if you use the "Parent" field to enter the name of an entity that doesn't exist, you will get an uncaught exception with a message such as "NSInlineString does not respond to selector reason".

Workaround: Don't set the name to an an entity that doesn't exist. After this happens, save your changes and exit EOModeler.

Reference: 72926

Problem: Dragging an EOEntity into a newly-added nib doesn't always work if you already have the EOModel in the project

Description: Dragging an EOEntity into a nib will fail if the nib was recently added to the project. The general sequence of events leading to this problem is as follows:

1. Open an existing project that already includes an EOModel.
2. Open an existing nib.
3. Create a new nib, and save it into the project.
4. Try to drag an entity out of the model into your nib.

Now Interface Builder will ask you if you want add the model to your project, even thought the model is already in the project. Regardless of your answer, you won't see the EODisplayGroup or EOEditingContext show up in your new nib.

The problem is that each nib file gets assigned a model group when it is first opened, so the new nib starts with an empty model group. The bug is that the model group isn't being updated when you save the nib into a project.

Workaround: Revert the document.

Reference: 72662

Problem: If your model has compound primary keys and uses inheritance, performing a consistency check will raise an exception.

Description: If you have a model with compound primary keys and you're using inheritance, running a consistency check will raise an exception, even if the model is fine. This is tied to the inheritance checking performed by a consistency check.

Workaround: You can disable inheritance checking. This is useful if you have compound primary keys and no inheritance. To do this, use the following command from a shell:

```
defaults write EOModeler DisableInheritenceCheck YES
```

Reference: 67804

Problem: EOModeler can't find model from another framework.

Description: Problem 1: When you load a nib file into your application, it can't find the model you have in your framework.
Problem 2: When you save changes to a model file in your framework, the change isn't reflected in open nib files.

Enterprise Objects Framework 2.0 finds models for an application by looking for all models in the application and in any frameworks used by the application. By default Enterprise Objects Framework looks for referenced framework in their *installation* location. Thus if you don't have the framework that contains the model installed, Enterprise Objects Framework won't find the model. Similarly, a change saved to a model in the framework project isn't noticed by Interface Builder since it's looking at the *installed* version, not the source version.

Workaround: You can tell Enterprise Objects Framework to look at the source for your framework projects by using the following defaults command (executed in a shell):

```
defaults write NSGlobalDomain EOProjectSourceSearchPath "($
(HOME)/myProjectsDirectory1, /myOtherProjectsDirectory)"
```

Then, when Interface Builder or EOModeler is looking for models contained in one of your frameworks, it will first search all project directories within **$(HOME)/myProjectsDirectory1** and **/myOtherProjectsDirectory** before searching for the built versions.

Reference: 64339

Problem: Derived attributes don't work properly.

Description: Sometimes when you're trying to create a derived attribute, you may encounter seemingly random errorsÐthe pop-up list in the Inspector reverting to Column, the text you typed in the field being erased or reverted, error panel telling you ªUnable to modify the attribute,º and so on.

Workaround: Always add a definition immediately after changing the attribute to from Column to Derived. Once you add a definition for a derived attribute, it can't be deletedÐyou can change it, but a definition must be present.

Reference: 62444

Problem: When you change a text field in an EOModeler Inspector and don't press Return, you can lose your changes.

Workaround: Always use the Return key.

Reference: 62447

Problem: Last column in EOModeler table mode isn't handled correctly when you resize.

Description: The last column in a table view can get clipped. If you widen the second-to-last column by dragging its right edge, the last column becomes narrower and the scroll knob doesn't appear until the mouse approaches the edge of the table view. If you save a model with a clipped column, then when you open it again the column remains clipped even if the browser is big enough to handle it.

Workaround: Widen the last column by dragging its right-most edge before widening the second-to-last column.

Reference: 62669

Problem: In EOModeler, the external types field isn't case insensitive for word-completion

Description: In the External Types field of the Attribute table view, word-completion needs to be be case-insensitive. For example, typing `v' should complete to `VARCHAR' just like typing `V' does.

Workaround: None.

Reference: 62727

Problem: Editing in Inspector with editing still active in table doesn't work.

Description: If you edit something in table mode, then, without tabbing out of the edited column, make a change in the Inspector, the change made in the table column is lost.

Workaround: None.

Reference: 63118

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

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

Workaround: Keep your table names shorter, or edit the SQL EOModeler generates to keep the constraint names shorter.

Reference: 63119

Problem: EOModeler generates bogus foreign key constraints for compound keys.

Description: Schema generation in EOModeler can produce some confused constraint definitions for Oracle and Informix if you have non-normalized data (such as the TALENT and TALENT_PHOTO tables).

Workaround: You can work around this by editing the SQL generation scripts, or by just not asking EOModeler to generate the constraint information.

Reference: 63169

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

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

Reference: 63183

Problem: Trying to generate schema SQL will raise if any of the primary key attributes don't have a column name.

Workaround: None.

Reference: 69386

Problem: Trying to drag a model or an entity from EOModeler into a nib file sometimes fails.

Description: If you try to drag a model or an entity from EOModeler into a nib file, Interface Builder asks if you would like to add this model to the project. Sometimes the attempt to add the model to the project will fail.

Workaround: Relaunch Interface Builder by double-clicking on the nib icon in Project Builder. Try dragging the EOModel or EOEntity nowÐeverything should work.

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 will cause invalid SQL to be generated. Definitions such as ªtitleº and ª0.0º will not work correctly. However, definitions such as ªatt1 + 5º should work correctly when ªatt1º specifies another attribute.

Workaround: None.

Reference: 57842

Problem: EODisplayGroups are missing their keys.

Description: Enterprise Objects Framework was probably unable to find the model file containing the entity used by the EODisplayGroup. To verify this, inspect the data source of the EODisplayGroup (click on the EODisplayGroup, bring up the Inspector panel, and selected the DataSource panel from the Inspector popup). If the ªModelº field is blank, then Enterprise Objects Framework was unable to find your model.

Workaround: Quit Interface Builder and verify that the model containing the entity referenced in the EODisplayGroup is added as a resource in the project containing the nib file.

On-line Examples

Reference: 73332

Problem: The Studios example has a bad hard-coded path in its framework search order that keeps it from building on NT.

Description: If you try to build the Studios example on NT, the build will fail due to a bad path in framework search path.

Workaround: To correct this problem, follow these steps:

1. Open the project in Project Builder.
2. Choose Tools -> Inspector.
3. In the Inspector's Build Attributes view, use the second pop-up menu to display
   Framework Search Order.
4. Remove **d:\NeXT\NextLibrary\Frameworks** from the search path list.

Reference: 73076

Problem: Oracle stored procedures example doesn't work

Description: The external names of the stored procedures are missing the prefix 'MOVIE_PROCEDURES.' This causes them to not work.

Workaround: Use the Stored Procedure view in EOModeler to change the external names of the two fetch procedures to MOVIE_PROCEDURES.FETCH_MOVIE_WITH_PRIMARY_KEY and MOVIE_PROCEDURES.FETCH_MOVIES_ALL.

Reference: 70557

Problem: The Customer example fails to propagate change to credit card.

Description: The relationship between CreditCard and Member is not modeled correctly in the examples. In the Customer application, you can modify a the number and type of a member's credit card, but the change to the primary key is not propagated to all of the objects that it relates to. Consequently, after saving the changes to the database, the credit card's number and type will have changed but the foreign key in the member table will still have the old values as the foreign key. This relationship would work correctly had the CreditCard class been modeled with an primary key identifier that was distinct from its number and type values and if the relationship were constructed on that primary key attribute. This would also be a more efficient implementation.

Workaround: None.

Reference: 69199

Problem: Error running **configure_examples** script on Windows NT.

Description: Running **configure_examples** will fail unless your path is properly specified.

Workaround: In order to install and run examples on NT, you need to add the following to your system path (in the System Control Panel):

```
.;c:\NeXT\LocalDeveloper\Executables
```

(The above assumes that $NEXT_ROOT is defined as **c:\NeXT**).

Reference: 69419

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

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.

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.