
WebObjects 5.2 Release Notes

(Legacy)

[Mac OS X Server > WebObjects](#)



2004-12-02



Apple Inc.
© 2003, 2004 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Logic, Mac, Mac OS, and WebObjects are trademarks of Apple Inc., registered in the United States and other countries.

Enterprise Objects is a trademark of Apple Inc.

NeXT is a trademark of NeXT Software, Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Chapter 1 **WebObjects 5.2 Release Notes** 5

Database Adaptors	5
Deployment	7
Development Tools	8
Direct to Web	10
Documentation	12
Enterprise JavaBeans	16
Enterprise Objects	17
Foundation Framework	21
Java Client	22
New Features	23
RapidDevelopment	24
Web Server Adaptors	24
WebObjects Framework	26
WebObjects	30
WOExamples	31
WOExtensions	32
WOJavaDoc	32
WOJPlot	34
WOJSPServlet	34

Document Revision History 37

WebObjects 5.2 Release Notes

Important: The information in this document is obsolete and should not be used for new development.

This document contains information about known issues and workarounds for this WebObjects release.

Database Adaptors

Radar #2897052

With `MSSQLServer`, `channel.describeModelWithTableNames(tableNames)` can throw an exception.

Description:

For `MSSQLServer`, `channel.describeTableNames()` will include 2 or 3 system tables. For older JDBC drivers for `MSSQLServer`, using this list for `channel.describeModelWithTableNames(tableNames)` will then throw an exception.

Workaround:

Use an up-to-date JDBC driver. This has been tested successfully with Microsoft SQL Server 2000 Driver for JDBC, Version 2.2.0022.

Radar #2918829

Key value coding fails to coerce the scale and/or precision of `BigDecimal` values. And when you try to save, the JDBC Adaptor throws an exception because of excessive scale.

Description:

Neither key value coding nor the JDBC adaptor performs any rounding; otherwise, information would get lost. Rounding could occur only if the underlying database/driver were to do so independently of EOF.

Workaround:

Please make sure that the scale and precision of a value (especially `BigDecimal`) does not exceed the scale and precision of the corresponding attribute in the model.

Radar #2924574

MySQL writes `NOW()` timestamp value instead of `NULL`.

Description:

According to the MySQL documentation:

Automatic updating of the first `TIMESTAMP` column occurs under any of the following conditions:

- * The column is not specified explicitly in an `INSERT` or `LOAD DATA INFILE` statement.
- * The column is not specified explicitly in an `UPDATE` statement and some other column changes value. (Note that an `UPDATE` that sets a column to the value it already has will not cause the `TIMESTAMP` column to be updated, because if you set a column to its current value, MySQL ignores the update for efficiency.)
- * You explicitly set the `TIMESTAMP` column to `NULL`.

On the initial insert of an EO with a null timestamp, EOF will explicitly insert the `NULL` which fires the trigger in the MySQL database, automatically setting the value to `NOW()`. That means you cannot reliably use the first timestamp column of a MySQL table for locking in EOF. Furthermore, since the first timestamp column will change for any update, your EO business logic has to be careful not to rely on that value so it hardly seems worth trying to use it as a class property. You can use the second (and greater) timestamp columns for locking if you first create a "dummy" (unused) timestamp column.

Workaround:

Avoid using the first timestamp column of a MySQL table as a class property and do not allow null on any MySQL timestamp columns in the model.

Radar #3050038

Executing a Sybase stored procedure causes an exception if it was created with the "unchained" transaction mode.

Description:

A Sybase stored procedure gets tagged as "chained" or "unchained" depending on the environment in which it was created. It will cause an exception if it is invoked within a different transaction mode. EOF always turns off autocommit, which means it always operates in what Sybase calls "chained" mode. Stored procedures declared in an "unchained" environment will not work with EOF. The error message will look like this:

```
SQL State:ZZZZZ -- error code: 7713 -- msg: Stored procedure 'MY_PROCEDURE' may be run only in unchained transaction mode. The 'SET CHAINED OFF' command will cause the current session to use unchained transaction mode.
```

The solution is to change the transaction mode of the stored procedure to either "chained" or "anymode".

Workaround:

Use `isql` to issue the following command (substituting your procedure name for `MY_PROCEDURE`):

```
sp_procxmode MY_PROCEDURE, anymode
```

Radar #3067643

The "Synchronize Schema" menu item in EOModeler is always disabled for models that use Sybase or Microsoft SQL Server.

Description:

The `SybasePlugin` and `MicrosoftPlugin` do not support the schema synchronization feature so the menu item is disabled in EOModeler.

Workaround:

Use database vendor tools (such as isql) to maintain your schema. Use the EOModeler command "New Updated Model" to update your model.

Radar #3069162

Sybase data-definition language (DDL) statements are not allowed inside a transaction.

Description:

EOF uses a transaction when creating tables and executing various data-definition language (DDL) statements. By default, Sybase databases do not allow DDL statements within a transaction. The user must explicitly configure his database to allow 'ddl in tran'.

Workaround:

Use isql to issue the following command (substituting your database name for MY_DB):

```
sp_dboption MY_DB, 'ddl in tran', true
```

Radar #3071404

The JDBC Adaptor generates incorrect SQL for MySQL outer joins.

Description:

By default, EOF uses a non-standard outer join operator "*"=" which is not supported by MySQL. The current EOF API does not support using the standard SQL-92 outer join syntax in the FROM clause. Other databases such as Oracle and Sybase have their own ways of specifying outer join operators, which are supported directly by the JDBC Adaptor.

Workaround:

None. Avoid defining relationships with outer joins in MySQL models.

Deployment

Radar #2282569

Applications with dynamic images may leak them.

Description:

If you are testing such an application with the playback tool, the tool will not do the necessary callbacks to retrieve dynamic images. Therefore, these images will stay in the data cache of the application and never be released. Same can happen once deployed, as clients disable image loading from their browser. You will see your application leaking when it is not really; it is just waiting indefinitely for someone to come ask for an image.

Workaround:

When using the `data, mimeType` bindings for dynamic images, try to use the key binding as well. With the key binding set to `employee.picture`, only once will the picture be saved in memory and stay there, limiting the amount of data leaked. Otherwise, you can always use the `-flushCache` call on `WOResourceManager` regularly. This clears out everything from the data cache, so this may impact currently connected users.

Development Tools

Radar #2180253

WebObjects Builder does not validate keys and values in the bindings inspector.

Description:

WebObjects Builder lets you write illegal values (such as values with unrecognized characters) for binding attributes and values. These values are written to your `.wod` file without validation, so your `.wod` file will not be correct.

Workaround:

Don't set your bindings to such values. If you do, close the component and reopen it. WebObjects Builder displays a parse error and puts you in source editing mode, where you can fix the corrupted bindings.

Radar #2181346

Centering several list items at once splits the list.

Description:

If you center a single list item, the list item properly places a paragraph around its text and centers the paragraph, leaving the list itself intact. However, if you select several list items and center them, the list will be split at the selection boundaries, and the selected pieces will be centered.

Workaround:

Center each list item independently.

Radar #2269932

Adding Java Client components (or an `EOJavaClientSubproject` in `ProjectBuilder` on Windows) to a project does not automatically add the necessary Java Client frameworks to the project.

Description:

If you create a project that was not intended to be used with Java Client in the beginning and add Java Client interface controllers and interface files (nib files) to it by hand, you also have to add all the frameworks needed by Java Client to the project by hand (for example `JavaEODistribution`, `JavaEOApplication`, `JavaEOInterface`, `JavaEOInterfaceSwing`).

Workaround:

Add the necessary Java Client frameworks (for example `JavaEODistribution`, `JavaEOApplication`, `JavaEOInterface`, `JavaEOInterfaceSwing`) to your project by hand.

Radar #2275826

EOF Interface Builder palettes won't parse keys from the source code of client-side Java classes.

Description:

If you drop a display group into a client-side nib, EOF Interface Builder won't recognize keys from the client-side class associated with the entity. Instead, it parses keys from the server side class.

Workaround:

Add any keys you need manually, using the EODisplayGroup inspector.

Radar #2278982

Opening components with very large tables causes WebObjects Builder to hang or crash.

Description:

A component containing a very large table (several thousand pixels by several thousand pixels, for example) will overtax WebObjects Builder, causing it to hang or crash.

Workaround:

None. Make the table smaller or edit the table with a machine that has more memory.

Radar #2418323

Subprojects deeper than two levels do not render HTML in rapid turnaround mode.

Description:

In development mode, rapid turnaround only works for the resource files in your top level project directory and in the first level of subprojects.

Workaround:

Add each of the subprojects' subprojects directory paths to the `NSProjectSearchPath` user default. See the documentation on user defaults to learn how to modify them.

Radar #2773063

Problems launching projects on Mac OS X when they were created on Windows.

Description:

File execution permissions are not created for the Unix launch script file (the one without the `.cmd`) or are not preserved when transferring a WebObjects application from Windows to a Unix machine.

Workaround:

`chmod a+x <launchScriptPath>` after the transfer and installation of the `.woa` is complete.

Radar #2841084

Fetch Specs which reference other entities generate incorrect SQL.

Description:

If the qualifier for a fetch spec references another entity (table) the generated SQL code is incorrect. While it contains the joins, the referenced tables are missing from the FROM clause.

Workaround:

Select "Use Raw SQL Expression" in the "SQL" pane and manually enter the SQL code.

Radar #3026896

Interface Builder on Windows shows "Error 44E7" error panel.

Description:

Dragging an EOEntity from EOModeler or performing "Test Interface" results in an error dialog with the text "Error 44E7". The EnterpriseObjects palette on Windows does not have access to the WebObjects JavaJDBCAdaptor.

Workaround:

You can still develop JavaClient interfaces with Interface Builder, but you cannot use the Interface Builder "Test Interface" functionality. Also, you may still drag and drop from EOModeler to Interface Builder JavaClient interfaces by simply dismissing the error dialog.

Radar #3063754

Deployment Installs from PB on Mac OS X are not supported.

Description:

WebObjects applications and frameworks cannot be installed from within the Project Builder IDE.

Workaround:

To install for deployment always build from the command line with both Deployment and WebServer build styles:

```
sudo pbxbuild install DSTROOT=/ -buildstyle Deployment
sudo pbxbuild install DSTROOT=/ -buildstyle WebServer
```

Direct to Web

Radar #2201189

DirectToWeb does not handle queries with custom data types.

Description:

If one of your attributes has a custom data type (i.e. not one of the built-in types such as NSNumber, NSDecimalNumber, NSDate, NSCalendateDate) you can get an exception when DirectToWeb is trying to display it.

Workaround:

Using the Web Assistant, navigate to the faulty query page in expert mode and hide the property that has a custom type.

Radar #2274611

DirectToWeb raises on custom or non-existent keys.

Description:

An EOUnknownKey exception can be raised when you run a DirectToWeb application. This behavior is due to either a mistyped key in the Web Assistant or the removal or renaming of an attribute or relationship in EOModeler.

Workaround:

Using the Web Assistant's expert mode, select the task/entity pair for which you're getting an exception and hide the key that is causing the problem.

Radar #2424172

Direct to Web applications can hang.

Description:

An editing context in your application may be locked and not properly unlocked. Due to the details of how Direct to Web locks editing contexts, an editing context isn't unlocked when you create a new Direct to Web page without rendering it.

In WebObjects, the locking and unlocking of `session.defaultEditingContext` is performed transparently. However, if your application uses other editing contexts, you need to lock them yourself.

Direct to Web helps you manage the locking of such editing contexts:

- The edit and inspect pages lock the editing context when the `setObject` method is invoked.
- The edit-relationship page locks the editing context when the `setObjectAndMasterRelationshipKey` method is invoked
- The edit-relationship page locks the editing context when the `setObjectAndMasterRelationshipKey` method is invoked.
- The inspect, edit, and edit-relationship pages lock the editing context of the object they manipulate when the `awake` method is invoked.
- The inspect, edit and edit-relationship pages unlock the editing context of the object they manipulate when the `sleep` method is invoked.

Therefore the following code:

```
myPage=D2W.factory().inspectPageForEntityNamed("F00",session());
myPage.setObject(myEO); return myPage;
```

requires no extra lock/unlock statements on your part, even when `myEO` is not in `session.defaultEditingContext`. However if you do not render the page right away but store it instead, the editing context remains locked unless you unlock it yourself.

Workaround:

Unlock the editing context explicitly by calling `sleep()`:

```
myPage=D2W.factory().inspectPageForEntityNamed("F00",session());
myPage.setObject(myEO); myPage.sleep();
thisOtherPage.setNextPage(myPage);
```

Radar #2507444

`DirectToWeb` does not find attributes in relationships that use inheritance.

Description:

Consider a model with an entity A that has a relationship to entity B. Also, entity B has a subclass called B1 which contains an attribute (`att1`) not present in entity B. Keypaths of the type `a.b.att1` will not work correctly with the rule system.

Workaround:

None, other than avoiding rules that contain the the keypath `a.b.att1` when you have the above scenario.

Documentation

Radar #2396118

`WOMessage` string encoding constants are strings, not integers.

Description:

In Java, character encodings are represented by strings, which is different from how character encodings were represented in previous versions of WebObjects. To determine the string that represents a character encoding supported by the JDK 1.3, consult the table at <http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html>

Workaround:

Specify the character encoding as a string.

Radar #2404574

Windows: `wotaskd` crashed and will not restart.

Description:

On Windows, if `wotaskd` crashes for some reason it should be restarted automatically by the `woservice` process. However, if the machine is set to have Dr Watson visual alerts appear in the event of a crash, the process will hang and `wotaskd` won't restart until the Dr Watson panel is dismissed.

Workaround:

Turn off Dr Watson visual alerts. This is usually required for any server machine.

Radar #2421581

The "cache-control: no-cache" http header causes Internet Explorer 5 to hang when using a proxy server.

Description:

By default WebObjects adds the "cache-control" header with a value of "no-cache" (among others) to responses. This particular header causes Microsoft's Internet Explorer 5 to hang if a proxy server is being used.

Workaround:

Set the `WOAllowsCacheControlHeader` user default to `NO`. This will cause the "cache-control" header (and all associated values) to be omitted from WebObjects responses.

Radar #2425256

On Windows 2000, some of the fonts used by tools such as ProjectBuilder, FileMerge, and WebObjectsBuilder are hard to read.

Description:

Windows 2000 introduced a new "message" font, Tahoma. This font is difficult to read at point sizes larger than 9.

Workaround:

Open the Display control panel, click on the Appearance tab, select Message Box from the Item pulldown menu, and then select another font (such as MS Sans Serif) instead of Tahoma.

Radar #2896355

Weblogic 7.0 breaks without `javaxml.jar` in its `jdk1.3.1` directory (Win2K/Sol).

Description:

In the WebObjects Windows 2000 and Solaris environment some errors have been observed with the Weblogic 7.0 Beta. When starting the server, errors occur when initializing the servlet adaptor.

Workaround:

Adding the `javaxml.jar` file to `bea/jdk1.3.1/jre/lib/ext` solves the problem. Also, adding the `javaxml.jar` file into an `ext` directory in `bea_home`, will not affect previous versions.

Radar #2995840

Many OpenBase processes appear in `top/Process Viewer/etc`.

Description:

The WebObjects Developer installation prestarts all the WebObjects example databases. These databases are always started at boot time.

Workaround:

Launch the OpenBase Manager application. One at a time, select each database for which you wish to disable auto start. Click the Stop button in the toolbar or use the Stop item in the Database menu. Once the database has been stopped (a red dot), select Configure Database from the toolbar (it looks like an 'i') or use `cmd-shift-C`. Uncheck the Start Database at Boot checkbox and click Set. Repeat for each undesirable example database.

Radar #3051044

When using the WOFileUpload dynamic element in a form, some of my other (non-file upload) form values are null.

Description:

In WebObjects 5.2, a new WOFileUpload API and behavior was introduced, allowing streamed uploads and downloads. As a result of this, some particular cases of WOFileUpload dynamic elements within a form may cause other form elements to not be retrievable. There are 2 cases: Direct Actions and WOFileUpload InputStream api. In both cases, because the uploaded file data isn't automatically streamed somewhere, any form values that are after the file data in the request cannot be retrieved until the file data has been read in. The WOFileUpload/inputStream api case is demonstrated in the FileUpload example ("Bad Component InputStream Upload in Form" link). In the Direct Action case, if non-file upload form values and file uploads are read out of order, the form values may not be accessible.

Workaround:

In a Direct Action, make sure that the form values and file uploads are read in order. In a WOFileUpload, if using the inputStream API, make sure that the WOFileUpload is the last element in the form.

Radar #3069079

Derived Attributes can generate incorrect SQL when fetching with joins.

Description:

The definition for a derived attribute specifies the literal SQL expression to use in the `SELECT` statement. Often this expression doesn't work correctly within the context of a join where table aliases are used (since the attribute definition does not use the appropriate table alias). A similar problem occurs when fetching raw rows using a derived attribute in a related entity.

Workaround:

There is no convenient workaround. In some cases, you may be able to use a column-based attribute and do some extra processing in Java code.

Radar #3071619

An exception occurs while serializing an EOEnterpriseObject.

Description:

If a subclass of EOCustomObject or EOGenericRecord has added non-static fields, it is responsible for conforming to the JDK Serializable contract. EOF automatically serializes all of an EO's class properties. However, the JDK serialization mechanism works on a per field basis in each subclass. Since this mechanism is unaware of the need to fire faults with `willRead()` or `willReadRelationship()`, fields in subclasses of EOEnterpriseObject can cause problems.

Workaround:

Declare the fields transient, or implement a `writeObject()` method to properly serialize the fields, invoking `willRead()` and `willReadRelationship()` when appropriate. As a general rule, any field which maps to a class property should be declared transient.

Radar #3072241

The WebServices third party library Axis does not support deserialization of cyclic graphs.

Description:

If one serializes out an object which represents cyclic graph like:

```
Vector a = new Vector();a.addElement("testing");  
a.addElement(a);
```

the current version of Axis (1.0) will create appropriate XML on the server side, but will be unable to deserialize it into objects on the client side. The EOEnterpriseObjectDeserializer and WOStringKeyMapDeserializer make efforts to work around this underlying problem. The deserializers provided by Axis, including all the basic Java types and JDK collections, will not support these object graphs.

Workaround:

A custom written deserializer installed over the default may be able to work around this issue on a per type basis.

Radar #3075525

Post-Installation Guide deprecated from WebObjects 5.2.

Description:

The WebObjects Post-Installation Guide has been deprecated from this release. Its contents have been incorporated into the Installation Guides and Deploying Applications book.

Workaround:

For any references to the Post-Installaton Guide, refer to the WebObjects 5.2 Installation Guide for Mac OS X, WebObjects 5.2 Installation Guide for Windows/Solaris, and Deploying Applications book.

Radar #3079530

Wrong path in the "Installation Guide for Windows and Solaris" document.

Description:

The WebObjects 5.2 Installation Guide for Windows and Solaris document says to mount the CD and navigate to the Deployment/SOLARIS/WebObjects/ directory to find the install.sh script. The actual mounted CD directory path is deployment/solaris.

Workaround:

Navigate to deployment/solaris to find the install.sh script.

Radar #3162526

Dynamic Elements Reference on WebObjects 5.2 CD missing files

Description:

The directories `/Developer/Documentation/Webobjects/Reference/DynamicElements/` installed from the WebObjects 5.2 CD are missing `.element` documentation for the classes `WOPopUpButton` through `WOVBScript`. This causes WebObjects Builder to give "documentation not available" errors when trying to browse documentation for these classes. Documentation for lexicographically earlier classes `WOActionURL` through `WOPasswordField` is complete, and browsing from the Builder works fine.

Workaround:

Access the Dynamic Elements reference from the website at <http://developer.apple.com/documentation/WebObjects/Reference/DynamicElements>.

Enterprise JavaBeans

Radar #2760239

Can't add more than one EJB to an EJB framework project using PBWO.

Description:

Due to architectural limitations of the assistant in PBWO, there is no assistant to add a bean to an existing project as there is for Project Builder on Mac OS X.

Workaround:

Use the assistant to create another EJB framework project, then add its resources to your project and merge the deployment descriptors manually. Or keep EJBs in separate frameworks.

Radar #2798695

An EJB framework titled "Beans", "Resources" or "Versions" will not compile.

Description:

An Enterprise Bean framework project with the names mentioned above will not compile successfully.

Workaround:

Don't use these names for your project.

Radar #2798740

EJB 1.0 jar files are not recognized by OpenEJBTool.

Description:

A jar file containing beans packaged according to the EJB 1.0 specification is not recognized by OpenEJBTool, which will throw an exception about the missing `META-INF/ejb-jar.xml` file.

Workaround:

Create the necessary files manually.

Enterprise Objects

Radar #2176526

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 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 toFighterPlanes.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`.

Radar #2280881

Concurrent multithreaded saves may bypass optimistic locking check.

Description:

Although it has not yet been verified, EOF may have a brief window where two sessions modifying the same object will get last-one-wins behavior, instead of a merge (which is still last-one-wins at the attribute level). For example:

- Editing context 1 locks, modifies object 1
- Editing context 2 locks, modifies object 1
- Editing context 1 saves changes, unlocks, change notification enqueued on editing context 1
- Editing context 2 saves its modifications to object 1 without merging in editing context 1's modifications.

Workaround:

None.

Radar #2329635

EOF can generate incorrect SQL for disjunctions of qualifiers where one of the qualifiers involves an optional relationship.

Description:

The form of the SQL generated for an EORQualifier is similar to the SQL generated for an EOAndQualifier, but with "OR" substituted for "AND". For example, the qualifier `attr1 = v1 OR rel.attr2 = v2` would generate a SQL WHERE clause something like:

```
WHERE (ENTITY.ATTR1 = v1 or REL.ATTR2 = v2) AND REL.PK = ENTITY.PK
```

For a mandatory relationship 'rel', there will always be a match in the REL table and the above SQL WHERE clause will end up giving the correct result. However, for an optional relationship with a NULL value, the WHERE clause fails to return any rows even if there were rows in the ENTITY table that matched the value v1. The appropriate SQL would be:

```
WHERE ENTITY.ATTR1 = v1 or (REL.ATTR2 = v2 AND REL.PK = ENTITY.PK)
```

which would give the correct results for all cases.

Workaround:

Separate your original fetch specification into independent fetch specifications that do not have disjunctions involving optional relationships, and concatenate the multiple results. Naturally, the same EO might be returned by multiple fetch specifications so you may need to filter out duplicates from the concatenated results. For complicated fetch specifications, it may be more efficient to use a custom SQL query (see documentation on `EOCustomQueryExpressionHintKey` in `EODatabaseContext.h`). If you choose to use custom SQL and you have multiple qualifiers across relationships, you should consider using `DISTINCT` since there may be multiple ways for one EO to satisfy the qualifier.

Radar #2359723

The EOF documentation implies that only a single SELECT statement is needed to perform a deep fetch when all the subentities of the fetch specification's entity map to the same table. However, only one specific case is handled with a single SELECT. In the general case, one SELECT is issued for each subentity.

Description:

EOF performs one SELECT for an inheritance hierarchy if each subentity is mapped to the same table and if each subentity either has a restricting qualifier of the form "attr = val", where "attr" is the same for all subentities and "val" is unique to each subentity, or the subentity is abstract and has no restricting qualifier. If some subentity doesn't qualify for the optimization, then its parent also doesn't qualify. However, if some subtree of an inheritance hierarchy qualifies for the single table fetch, then that subtree is fetched in one SELECT even though other subentities of the parent entity have to be fetched individually.

Workaround:

Arrange your single table inheritance heirarchy so that it meets the optimization criteria.

Radar #2431180

After calling `myEODatabaseContext.invalidateAllObjects()`, subsequent saves either fail to save changes or just raise.

Description:

If an application has any `EOEditingContexts` that contain unsaved inserted objects, any call to `myEODatabaseContext.invalidateAllObjects()` will leave those `EOEditingContexts` in an inconsistent state. If you try to save the changes in any such `EOEditingContext`, the inserted objects will not be saved, or you might get an exception because these objects are mistakenly treated as updated objects.

Workaround:

Replace all invocations of `dbc.invalidateAllObjects()`

with `dbc.invalidateObjectsWithGlobalIDs(dbc.database().snapshots().allKeys())`

Radar #2759733

The JDBC Adaptor does not "quote" SQL identifiers.

Description:

Any unusual SQL identifier (for example, a column name containing a space) needs to be quoted in an SQL statement. The `JDBCPlugIns` provided with the JDBC Adaptor do not add the required quotes. This can lead to parse errors from the database when these unusual identifiers are used in an SQL statement generated by the JDBC Adaptor.

Workaround:

If possible, change the SQL identifiers in your database schema so that quoting is not required. Otherwise, create a custom `JDBCPlugIn` subclass and turn on identifier quoting by setting the protected variable `_externalQuoteChar = null` in your plugin's constructor. This will cause `JDBCExpression`'s implementation of `externalNameQuoteCharacter()` to use the database's quote character. Note that `_externalQuoteChar` is type `String` (not `char`).

Radar #2792055

Additions and removals in a to-many relationship are not saved if the foreign key isn't marked as a class property or as used-for-locking, and also if there is no inverse relationship defined.

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. However, the problem will not occur if there is a inverse relationship defined for the to-many.

Workaround:

The simplest work-around is to mark attributes that are destination keys of a to-many relationship as used-for-locking. Alternatively, it is sufficient to define an inverse relationship for the to-many relationship. The inverse relationship does not need to be a class property. The final alternative is to make a small change to your application. At the beginning of the application (when all models are loaded into the model group), call `EOModelGroup.defaultGroup().loadAllModelObjects()`.

This one-liner will insure that all (hidden) inverse relationships are created before the first fetch operation, which in turn will cause the necessary foreign keys to be fetched (and then saved).

Radar #2854101

EOF does not generate proper SQL for horizontal inheritance when qualifying over a key path.

Description:

When fetching with an `EOFetchSpecification` containing an `EOKeyValueQualifier` whose key is a keypath to an entity (`SuperEntity`) for which there is a subentity (`SubEntity`) inheriting from it using horizontal inheritance, EOF only joins with `SuperEntity` and ignores `SubEntity`. In this situation, an incomplete result may be returned. Vertical and Single-Table inheritance are not affected by this bug.

Workaround:

None. Try to use single-table inheritance instead of horizontal inheritance.

Radar #2945851

When trying to promote a raw row to an EO, a `NullPointerException` is thrown.

Description:

When raw rows are fetched from an Oracle database using `EOUtilities.rawRowsForSQL()`, the dictionaries returned contain key names in all uppercase. Since Oracle represents everything in uppercase, these keys correspond to the column names and not the usual EOF convention.

Workaround:

You can control the labels used in returning the result set by specifying them explicitly in the SELECT list in your SQL statement. Make the label for the primary key column match your PK attribute name in your model. That way, `describeResults()` will use the label as key. Something like this should work...

```
SELECT FOO_ID "fooID", BAR, BAZ FROM MY_TABLE WHERE ...
```

Radar #2982434

A model should not have multiple class properties that refer to the same storage.

Description:

If the flattened property chain contains class properties, the flattened property should not also be a class property.

Workaround:

Do not mark the flattened property as a class property. Implement business logic methods that use key-value-coding expressions instead, for example "rel1.rel2.rel3" instead of "flattenedRel3".

Radar #3019437

Validation exceptions in EOF handle the undo stack inconsistently.

Description:

While performing an EOF operation (such as `saveChanges`) on an `EOEditingContext` which supports undo/redo (active by default), the state of the undo stack will differ depending on the kind of validation exception which is caught. For most validation exceptions, the undo stack will be untouched, and the application can invoke `undo()` or make some changes and try to save again.

However, for any validation exception stemming from delete propagation, `undo()` will already have been invoked. If an application wishes to try again, it will either need to apply the entire delete operation again, or `redo()` and then make a correction.

In generation, the only way to differentiate between the kinds of validation exceptions and which may require an `redo()` is to parse the exception message text.

Workaround:

None.

Foundation Framework

Radar #2926212

Java system properties that are defined but empty are not passed from the command line to the WebObjects application.

Description:

If you define an empty Java system property (for example, `-DMyProperty`) in the command line--on a terminal window, in Project Builder, or in ProjectBuilderWO--that definition is not passed to your application. This is due to ambiguity between NeXT-style property syntax (`-DNeXTProperty NeXTValue`) and an empty Java property definition in the command line.

Workaround:

Since WebObjects applications can read Java system properties from the application's Properties file, you can define empty properties there. The properties must be stored in a file named Properties in the application or framework's Resources directory. You can add a Properties file to the application or framework by adding

it to the Resources group in Project Builder. The Properties file must be in the format specified by `java.io.Properties`. In the unlikely event that the property needs to be visible before the `main` method of the Application class, you have to use the `JVMOptions` hook provided in the classpath file. For example, on the Mac OS X platform, this file is located at `path_to_your_app/app.woa/Contents/MacOS/MacOSClassPath.txt`. Anything to the right of the equals sign on the `JVMOptions` line will be inserted such that the JVM itself is able to find the property.

Radar #3005482

The string produced by the `toString()` method for JavaFoundation collection objects (e.g., `NSArray`, `NSDictionary`) doesn't serialize the object.

Description:

According to Java convention, `toString()` is just used for debugging purposes, not for serialization of objects. Consequently, the string produced by `toString()` on JavaFoundation collection objects will not necessarily produce a valid property list.

Workaround:

The collection classes implement `java.io.Serializable`, which is the supported way of serializing and deserializing the collection objects.

Radar #3063536

`WOXMLDecoder` will call the getter methods, if they are present, of your class during decoding. And the setter methods will not get called if the property's value has not changed.

Description:

As part of a bug fix and an optimization, `WOXMLDecoder` (with or without a mapping file) will now call `NSValidation.Utility.validateTakeValueForKeyPath()` instead of `NSKeyValueCodingAdditions.Utility.takeValueForKeyPath()`. There are two side effects of this change:

1. `validateTakeValueForKeyPath()` will end up calling the getter method for each property to be set. This is to compare if the value has changed. Thus if you have codes in your getter method that don't simply return the value, you might have unexpected results.
2. A consequence of (1) is that the setter method might not be called if the property's value has not changed.

Workaround:

Scan the getter methods for potential unexpected results. If the setter methods contain initialization codes, move them out to a separate method and call that method after the object has been decoded.

Java Client

Radar #3053769

Master/detail user interfaces do not necessarily delete objects.

Description:

In a master/detail user interface, the delete operation only removes objects from the relationship, they do not get deleted directly by the operation triggered through the display group/detail data source. Only if the relationship is set to Owns Destination, the objects also get deleted when removing them from the relationship.

This behavior can result in situations where objects that got temporarily inserted and removed afterwards are not visible in the user interface any more (and thus cannot be deleted), but cause validation errors during saves which prevent the save from succeeding.

Workaround:

Check whether your model should use the Owns Destination flag. If not, use a different kind of user interface, one where you have a separate display group and user interface for the object to insert and another one with an action insertion association to attach objects to the relationship. Or implement your own application logic (typically in an EOController subclass) to make sure objects really get deleted.

Radar #3149057

Cannot use shared EC on client side.

Description:

Using ESharedEditingContext with JavaClient shows inconsistent behavior. This means that a client shared EC won't automatically be loaded from a server shared EC. Also, a client shared EC won't automatically be loaded when an entity's objects are marked as shared in an EOModel. But explicitly fetching objects into a client shared EC should mean that a shared EC's functionality will work on the client.

Workaround:

Explicitly fetch objects into a client shared EC to ensure expected shared EC functionality on the client.

New Features

Radar #2256367

Configuration of WODisplayGroups in WebObjects Builder is not undoable.

Description:

WebObjects Builder won't undo any changes made to a WODisplayGroup via the configuration panel.

Workaround:

Changes can be reverted before dismissing the panel. Otherwise, you must restore the desired settings by hand.

RapidDevelopment

Radar #2809571

How do I find the association of a Cocoa/EOF or JavaClient UI widget?

Description:

If you need to identify the EOAssociation instance that manages a UI widget, you cannot rely on the widget's delegate or event listener being the EOAssociation instance.

Workaround:

1. Subclass the `EOWidgetAssociation Plugin` class related to the widget of interest.
2. In the plug-in's constructor, store the association in a dictionary with the widget as the key.
3. When you need information on the association of a widget, look for the association in the dictionary.
4. Invoke `com.webobjects.eointerface.EOWidgetPluginRegistry.registerWidgetPluginClass` to make your plugin the default plugin.

Example:

```
public class MyCocoaTextFieldPlugin extends EOCocoaTextFieldPlugin {
    public MyCocoaTextFieldPlugin(EOWidgetAssociation association, Object widget)
    {
        super(association, widget);
        MyUtility.myStaticMutableDictionary().setObjectForKey(association, new
        Integer(widget.hashCode()));
    }
}
```

in the NSApplication delegate's static initializer, invoke

```
com.webobjects.eointerface.EOWidgetPluginRegistry.registerWidgetPluginClass(EOTextAssociation.class,
NSTextField.class, MyCocoaTextFieldPlugin.class); Later on, you can call EOWidgetAssociation
association = MyUtility.myStaticMutableDictionary().objectForKey(new
Integer(widgetFromNotification.hashCode()));
```

Web Server Adaptors

Radar #2454840

When using the CGI WebObjects HTTP Adaptor in multicast mode, performance is suboptimal.

Description:

Configuring the CGI adaptor to use multicast to discover wotaskd daemons on the deployment subnet results in very slow performance. Configuring to use non-multicast mode and to already know which wotaskd daemons it should talk to results in a seven-fold performance increase. CGI in multicast mode is the configuration that will generate the most noise on your deployment subnet, so it is not recommended except for development or functionality testing.

Workaround:

If CGI performance is unacceptable in multicast mode, then one of the non-multicast modes should be used. Note that multicast is not the default mode, so you should already know how to switch the CGI back to a non-multicast mode. More information is available in the WebObjects Deployment guide.

Radar #2455171

WOHTTPConnections cannot be shared between worker threads.

Description:

Due to the fact that WOHTTPConnection currently has one input buffer per instance, and no locking thereof, WOHTTPConnection objects cannot easily be shared between threads, as unpredictable results will occur.

Workaround:

Either avoid sharing WOHTTPConnection objects between threads, or add additional locking logic in your own code.

Radar #2766065

When using a WebObjects application running in WebLogic, you cannot connect to WebObjects application deployed through the WebObjects HTTP adaptor.

Description:

WebLogic can be configured in 2 ways. The first way is by MIME type, and the second is by Path. However, there seem to be some issues with configuration by Path.

Workaround:

If WebLogic can be configured to service URLs by path, a standard WebObjects deployment should be able to work concurrently with a WebLogic deployment.

Radar #3046661

Solaris HTTP adaptor must use same compiler as web server.

Description:

The WebObjects 5.2 HTTP adaptor binaries for Solaris were compiled using gcc 3.1. If a customer compiles their own version of a the web server using a different compiler (the Sun compiler, for example), the prebuilt plugin adaptor binary may be incompatible.

Workaround:

Build the HTTP adaptor using the same compiler that was used to build the web server.

WebObjects Framework

Radar #2260519

`WOApplication.application().port()` returns a nonsensical result.

Description:

If your user defaults have `WOPort=-1`, then `WOApplication.application().port()` returns `-1`.

Workaround:

Access the port number directly from the adaptor via:

```
WOApplication.application().adaptors().objectAtIndex(0).port()
```

Radar #2273821

The session size numbers on the `WOStats` page appear to be too large.

Description:

The session size numbers are only an approximation of the actual size of the session. When an application has finished initializing, its memory size is noted. The session size that is reported is based on the memory increase since startup, and is divided by the current number of active sessions. The memory increase includes any cached data and/or data initialized after startup, skewing the size computation.

Workaround:

None.

Radar #2274039

IDs in Cookies are dropped at the front door of applications.

Description:

When "IDs in Cookies" is turned on for a WebObjects application, and you enter that application through the default request handler, cookies with old IDs are ignored in the request. In the response, new cookies for new IDs are sent back, or, in the case when there wouldn't be any new ID needed (such as in a stateless application), cookies clearing out the old IDs are sent back. This ensures that a user gets a new session when reentering a site, even if the user's browser keeps old, non-expired cookies around. This behavior is non-customizable.

Workaround:

Use the API `shouldRestoreSessionOnCleanEntry(WORequest aRequest)` on the `WOApplication` class to control whether cookie sessions are dropped as above.

Radar #2276414

Sessions do not time out with `WOAdditionalAdaptors`.

Description:

If you attempt to use the WOAdditionalAdaptors feature, sessions may not timeout.

Workaround:

None.

Radar #2282001

The current context and the current request aren't accessible from within a WOSession initializer.

Description:

WOSession's initializer doesn't set its context instance variable, so it isn't possible to access either the current context or the current request for use in your WOSession subclass initializer.

Workaround:

There are two ways to work around this problem. The simplest is to implement `awake()` on your session. At this point, the context instance variable is set. Since `awake()` is invoked for every request, though, you'll have to protect your code so it is executed only once in `awake()` (if it is initialization code, it most likely needs to happen only once). A better solution, assuming you only need the request object, is to implement `createSessionForRequest()` in your subclass of `WOApplication`:

```
public WOSession createSessionForRequest(WORequest aRequest) {
    Session session = (Session)super.createSessionForRequest(aRequest)// put
your custom initialization that needs the request object here.;
    return session;
}
```

Radar #2290990

Standard error and warning messages aren't localized.

Description:

When running a WebObjects application, some standard error and warning messages--such as "You have backtracked too far"--can be sent directly to the user. These messages aren't localized and can confuse a non-English user.

Workaround:

Provide your own implementation for the different handle...

Error methods in `WOApplication:handlePageRestorationError` for localizing the "backtracked too far" messages
`handleSessionCreationError` when the application is unable to create a new Session
`handleSessionRestorationError` when the `WOApplication` is unable to find the Session

It's always a good idea to implement the above methods, providing your own logic and user interface to handle exceptional conditions.

Radar #2372492

Cookie expire header date format change.

Description:

In order to comply with RFC 2109 and the Netscape informal specification, the date format used with the cookie expire header has been changed to "%a, %d-%b-%Y, %H:%M:%S GMT". Previously %A was used rather than %a. %a formats weekdays as Mon, Tue, Wed, etc. rather than the full weekday name.

Workaround:

None

Radar #2425444

When binding a date object to a WOTextField, you must also bind a formatter.

Description:

If you bind an NSTimestamp object to a WOTextField's value attribute but do not bind either a date format string to `dateFormat` or a formatter object to `formatter`, after submission the date object will be invalid. For example, entering '10/10/20' will store an NSTimestamp with a `yearOfCommonEra`: 1.

Workaround:

Bind either the `dateFormat` or `formatter` attributes as well as the value attribute.

Radar #2574754

WOXMLDecoder cannot deserialize non-EO to-one relationships back from the XML document that is serialized by WOXMLCoder; the inverse relationship is missing.

Description:

If you use WOXMLCoder to serialize an EO that has a to-one relationship, WOXMLDecoder deserializes the same to-one relationship back. The same is not true for a non-EO graph that contains a to-one relationship. For example, assume that object A contains object B, and object B contains object A. The serialized object graph might look like this

```
<A_tag type="A" objectID="1"> <BinsideA type="B" objectID="2">
<AinsideB type="A" objectIDRef="1"></AinsideB> </BinsideA></A_tag>
```

When you use WOXMLDecoder to deserialize the XML document, object B contains `null` instead of object A.

Workaround:

Use NSXMLOutputStream and NSXMLInputStream. Alternatively, you can implement post-construction initializers. For example, implement the `setA` method in class B and invoke it after object A has been constructed.

Radar #2818115

Unable to send E-mail in Japanese using WOMailDelivery.

Description:

WOMailDelivery was implemented using `sun.net.smtp.SmtpClient`, which is no long supported by Sun. This class does not support alternate encodings.

Workaround:

Subclass `WOMailDelivery`, and override the `sendEmail` method to use the `JavaMail` API instead of `SmtplibClient`. Create a static method in your class to return an allocated, shared instance.

Radar #2985159

Nasty exception inexplicably thrown by `WOHTMLWebObjectTag` for a component that seems OK.

Description:

The declaration parser in `WebObjects` allows users to comment out sections of their declarations file (named ending in `.wod`). This commenting behavior supports both old (`/**/`) and new (`/*!`) style comments. Unfortunately, such behavior conflicts with attempts to parse bindings like: `accept = "*/"`;

Workaround:

Make the binding in question to a method or field of type `java.lang.String` that provides the value `"*/"`.

Radar #3005696

`request.addCookie(new WOCookie())` and `request.setHeaders(new NSMutableArray(new WOCookie().headerString()), "cookie")` result in different `Cookie` value strings.

Description:

`request.addCookie(new WOCookie(aName, aValue))` will result in a header of the form `Cookie: aName=aValue` being added to the request.

```
request.setHeaders(new NSMutableArray(new WOCookie(aName, aValue).headerString()),
"cookie")
```

will result in a header of the form `Cookie: aName=aValue; version=1` being added to the request.

This is because `WOCookie.headerString()` generates a string intended for use in a `WOResponse`. If you wish to use cookies on a `WORequest`, you must either use the `addCookie()` API or manually create your header value string.

Workaround:

Use `request.addCookie()`.

Radar #3056127

You cannot upload files with a size greater than 2GB.

Description:

Because of underlying implementation details, you cannot upload files with a size greater than 2GB. Attempting to do so will result in undefined behavior.

Workaround:

None.

WebObjects

Radar #2806377

Fatal exception when programmatically creating an EOEditingContext in a Cocoa/EOF java class.

Description:

In some circumstances, creating an EOEditingContext programmatically in a Cocoa/EOF java class throws a `NullPointerException`. Specifically when the editing context is created very early in the application startup phase, such as in the `NSApplication` delegate's constructor or `applicationDidFinishLaunching` delegate method AND there is no EOEditingContext instantiated in the related Nib file.

Workaround:

Leave an EOEditingContext instance in the application's Nib files (One is added to the Nib for you by the WOAssistant).ORInvoke `com.webobjects.eointerface.cocoa.EOCocoaUtilities.registerCocoaSet()` before programmatically creating an EOEditingContext

Radar #2907225

JavaClient applications using WebStart(tm) against a Windows 2000 Professional server may fail to launch with a Download Error.

Description:

If the server is running Windows 2000 Professional and IIS 5.x, JavaClient applications may fail to launch through WebStart(tm). WebStart will return a Download Error - "Unable to load resource."

This is because IIS running on Windows 2000 Professional is limited to 10 concurrent requests.

Workaround:

Upgrade to Windows 2000 Server, or try loading the JavaClient application again.

Radar #2948731

Deadlock with EOSharedEditingContext.

Description:

A race condition can cause an application that uses EOSharedEditingContext to deadlock. This deadlock can occur when the shared editing context receives a change notification concurrent with access from another thread. The deadlock can be identified by thread(s) being blocked with the following appearing in the backtrace:

```
ava.lang.Object.wait (Native Method)
java.lang.Object.wait (Object.java:420)
com.webobjects.foundation.NSMultiReaderLock$ConditionLock.await
(NSMultiReaderLock.java:503)
```

Workaround:

For more details, see What's New in WebObjects 5.2 and API Changes from 5.1.

Radar #3035679

Top-level components can't be stateless.

Description:

To make a component stateless, one must override the component's `isStateless` method so that it returns true. However, overriding this method on a top-level component (i.e. `Main`) won't have any effect on his stateless status. In fact, a new top-level component will be replicated as needed instead of being shared between sessions. The stateless status was designed to be used for nested subcomponents because it is there where the advantage of sharing one instance has more relevance.

Workaround:

Not applicable.

WOExamples

Radar #3001491

Your application doesn't respond at all after it's started up. Only occurs on Mac OS X.

Description:

The symptoms typically seen are an error page that contains text like:

```
Error: com.webobjects.appserver.WOPageNotFoundException:  
Unable to create page 'Main'.Reason:  
<webobjectsexamples.thinkmovies.Application>  
Unable to create page 'Main'.
```

In the stack trace that follows, you should notice no references to your `WOApplication` subclass, either (which is typically named "Application").

The lack of your `WOApplication` subclass and your "Main" `WOComponent` subclass should indicate to you that the main bundle (i.e. the `.woa`) is not being recognized by WebObjects. If you have a symbolic link that points to your `.woa` directory and you're using that to access the launch script (the executable file that is the same name as your `.woa` directory with the `.woa` extension stripped from it), check whether the path to which this symbolic link points ends in one or more slash characters (`/`).

There's a bug in the Java Runtime Environment that shipped with Mac OS X 10.2 that prevents `java.io.File.getCanonicalPath()` from properly removing slashes from the end of the paths that it returns, and this problem confuses part of the WebObjects runtime.

Workaround:

Recreate the symbolic link in question, and exclude slash characters from the end of the path to which the recreated symbolic link points.

WOExtensions

Radar #2395684

JSValidatedField WOExtensions component fails to check the validity of fields if the user does not tab through them.

Description:

The Java Script code for JSValidatedField is only invoked when the user tabs through the field. If the user submits the form without tabbing through the fields, the fields will not be validated.

Workaround:

None.

Radar #3077580

An exception page provides stale information.

Description:

WOExceptionPage in JavaWOExtensions.framework provides optional communication between the WebObjects application and Project Builder or PBWO. This communication allows users to cause PB or PBWO to display the source code for a line from the exception's stack trace. Only projects available to the user, such as their application project, may be displayed in this way. To avoid performance issues when invoking the application actions underlying this communication, WebObjects does not prevent the browser from caching the exception page. This can cause a stale page to be displayed by the browser, depending upon how a given browser caches pages.

Workaround:

Quit and relaunch the browser. More simply, you may refresh the page in the browser (e.g., by hitting the Refresh button while the stale page is displayed).

WOJavaDoc

Radar #2370377

Locking operations aren't supported in the client side of a Java Client application.

Description:

Locking is not implemented in Java Client's distribution layer (especially EODistributedObjectStore), but there is public API in those classes that makes it appear otherwise. The EODistributedObjectStore method isObjectLockedWithGlobalID always returns false because locking isn't supported in the client. Similarly, the EODistributedObjectStore method lockObjectWithGlobalID raises an exception. The EOEditingContext locking methods (lockObjectWithGlobalID, lockObject, locksObjectBeforeFirstModification, setLocksObjectBeforeFirstModification, and isObjectLockedWithGlobalID) cannot be used on the client side; lockObjectWithGlobalID and

`lockObject` raise on first use; `locksObjectBeforeFirstModification` and `setLocksObjectBeforeFirstModification` trigger a raise if set to true and you modify an object; and `isObjectLockedWithGlobalID` always returns false.

Workaround:

None.

Radar #2574946

WOXMLCoding has an additional method.

Description:

There is a new `classForCoder()` method in the WOXMLCoding interface that is the same as the `classForCoder()` method in the NSCoding interface. The WOXMLCoder uses the method to specify the type attribute of an encoded object element.

For the classes with built-in support from the WOXMLCoder that implement the NSCoding interface, the `classForCoder()` method that already exists gets used.

Workaround:

N/A

Radar #2855641

In-memory qualification and sorting differs slightly from SQL.

Description:

EOQualifiers are generally used to generate SQL, but they also can be used for "in-memory" qualification, where the results are calculated directly. Due to differences in the semantics between Java and SQL, there may be edge cases where the results can differ, particularly if null values are involved. This release note describes how in-memory qualification treats nulls.

From EOQualifier's point of view, nothing is equal to null. However, `isEqual` or `isNotEqual` will treat `null == null`. Other operators such as `<`, `<=`, `>`, `=>`, etc. all return false if either operand is null. That is, `null == null`, but `!(null <= null)`.

This behavior is intended to be as close to SQL as possible. For `EOSortOrdering`, null is essentially treated as negative infinity. An array with null (or `NSKeyValueCoding.NullValue`) will have those elements sorted toward the very beginning if the sort selector is `CompareAscending` or at the very end for `CompareDescending`.

Workaround:

None.

WOJPlot

Radar #3002407

WebObjects Applications that use JavaPlot won't work unless a user is logged in.

Description:

Because the JDK objects that JavaPlot uses for image rendering require AWT and a Window Server to be running, any application drawing graphs with JavaPlot will require a user to be logged in. This is true up to and including Java 1.3

Workaround:

Make sure a user is logged in. There is currently no alternative, as Sun will not provide a patch for Java 1.3 . It is fixed in Java 1.4 . For more information, please see the following Sun bug reports:

<http://developer.java.sun.com/developer/bugParade/bugs/4038225.html>

<http://developer.java.sun.com/developer/bugParade/bugs/4281163.html>

WOJSPServlet

Radar #3038197

Potential performance problem when deploying WebObjects applications on Tomcat 4.0.x, integrating with Apache WebServer using ajp13/WARP protocols.

Description:

In general, WebObjects applications are deployed in Tomcat 4.0.x servlet containers without performance problems. We have tested this either running Tomcat 4.0.x in a standalone mode or integrated with Apache WebServer. There is a particular edge case that you might perceive delay in the response.

1. You are deploying on Tomcat 4.0.x, integrating with Apache WebServer.
2. The connector/protocol that you have chosen for the integration is `mod_jk/ajp13` or `mod_webapp/WARP` (`mod_jk/ajp12` doesn't have this edge case problem but it is not recommended in general by Apache. See Apache's website for more details)
3. Your WO application has extremely fast response < 200ms (Usually this means you have little or no database activity). And the content is small, usually smaller than 1500 bytes.
4. You are deploying in a fast LAN.
5. Your Tomcat 4.0.x container runs on Mac OS X.

If all the conditions are satisfied, you will see that your response will take at least 200ms per request-response loop. Here is a short description of the source of the problem for ajp13 protocol.

The way the protocol was designed is that the Tomcat container will respond in 3 chunks: a chunk of short headers, a chunk of body, and a short chunk of end-response content. Because `TCP_NODELAY` is turned off by default on Mac OS X, chunks do not get sent to the Apache WebServer until either an `ack` is received or the chunk size is big enough. Tomcat 4.0.x should have turn on `TCP_NODELAY` when using `ajp13` protocol.

Workaround:

There are two options.

1. You can simply ignore this problem since 200ms is not a lot for your users when they are using the browser, especially on the Internet.
2. If you are doing stress testing and you (or your management) are bothered by the 200ms delay, you can patch Tomcat's connectors. Here we describe the patch for `ajp13`.
 - a. You have to get the source codes for Tomcat 4.0.x connectors and try to compile the JAR for `tomcat-ajp.jar`.
 - b. Assuming that you are using the class `org.apache.ajp.tomcat4.Ajp13Connector` (the default) as your `ajp13` connector, you open up `org/apache/ajp/Ajp13.java` in your favorite editor and search for the method `setSocket()`.
 - c. Insert `socket.setTcpNoDelay(true)` after the `setSoLinger()` call.
 - d. Follow the documentation on compiling the Java side of the `ajp13` connector. You can ignore the C side (the Apache part). One of the JARs that you should get is `tomcat-ajp.jar`.
 - e. Copy `tomcat-ajp.jar` to `<TOMCAT_DIR>/server/lib`, after you have backed up the original.

Radar #3072947

Some WebObjects applications (particularly `JavaClient` or `DirectToJavaClient`) may fail to install as Servlet Single Directory Deployments (SSDD) in WebSphere on Windows.

Description:

Windows has a total file path limit of 255 characters. Normally this is sufficient, however a combination of Project Builder (for MacOSX) frameworks and SSDD can cause some files to exceed this file path limit. The problem is excaerbated by the process of converting symlinks to Windows; because symlinks are flattened, the Project Builder (for MacOSX) build frameworks have an internal structure of `/Versions/Current/...` as well as `Resources` and `WebServerResources`.

Workaround:

In many cases, simply deleting the top level `Versions` directory in any framework build by Project Builder (for MacOSX) is sufficient. As long as the top level `Resources` and `WebServerResources` folders remain, the app should run fine. In some circumstances, the application classpath (in the `web.xml`) may need to be tweaked if it referenced the framework jar as `...framework/Versions/A/Resources/...`

Document Revision History

This table describes the changes to *WebObjects 5.2 Release Notes*.

Date	Notes
2004-12-02	Remove text causing broken links.
2003-06-12	Updated with bug fixes for WWDC 2003.
2002-10-01	Released with WebObjects 5.2 release.

REVISION HISTORY

Document Revision History