Installation Instructions and Release Notes

WebObjects 3.5.1 Patch 4

September 16, 1999

NXT-6059

 Apple Computer, Inc.

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for printing or clerical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014-2084
408-996-1010
http://www.apple.com/

Introduction

This document describes Apple's Patch 4 for WebObjects 3.5.1. This patch set includes fixes that address the Year 2000 compliance issue (century rollover). Apple recommends that all users of WebObjects install the appropriate patch set. These Release Notes also document the issues found in Enterprise Objects Framework (EOF).

What This Patch Covers

This patch consists of patch binaries for Solaris, Windows NT, and HP-UX platforms, plus descriptions of additional steps required for full compliance on developer installations for all platforms. This patch affects the development and deployment versions of WebObjects 3.5.1 on Windows NT, Solaris, and HP-UX. No WebObjects patch is required for users of WebObjects 3.5.1 on OPENSTEP 4.2 for Mach; however, you must have OPENSTEP 4.2 for Mach Patch 4 or later installed on your OPENSTEP system. WebObjects users on OPENSTEP for Mach should also read sections 3 and 4 of this document; these sections describe some manual steps you should take to avoid some Year 2000-related problems.

Apple has tested WebObjects 3.5.1 with Patch 4 installed and has determined based on the results that the software meets Apple's standards of Year 2000 compliance*. This includes the use of the database adaptors included with EOF, when used through EOF. It has been determined that the direct use of client library API, custom SQL statements, or other use of the database adaptors or software other than through the EOF APIs may produce results that do not meet Apple's standard of Year 2000 compliance.

** For Apple, Year 2000 compliance means that the software does not produce errors processing date data in connection with the year change from December 31, 1999 to January 1, 2000, when using accurate date data in accordance with its documentation, provided all other products used with it properly exchange date data with the Enterprise Software product.*

What Is Not Covered

Third Party Software Applications

WebObjects 3.5.1 is designed to support certain third-party products, including but not limited to databases, web servers and HTML editors. Apple has not tested the Year 2000 status of these products and will not guarantee or provide support for them.

Operating System Software

WebObjects 3.5.1 can be used under OPENSTEP for Mach, Windows NT, Solaris, and HP-UX. Because WebObjects depends on the underlying operating system for some tine and date information, Apple cannot guarantee the operation of WebObjects after December 31, 1999 unless the underlying operating system is free of Year 2000 problems. Please contact your operating system vendor regarding the Year 2000 status of your system. WebObjects installations running on OPENSTEP 4.2 for Mach systems must have OPENSTEP 4.2 for Mach Patch 4 or a later patch version installed on their system for Year 2000 compliance.

About this Document

This document is divided into the following sections:

Section 1: Installation Notes and Instructions

Section 2: Issues - WebObjects 3.5.1 Patch 4

Section 3: Issues - WebObjects Examples (all platforms)

Section 4: Issues - Enterprise Objects Framework

Section 1: Installation Notes and Instructions

WebObjects 3.5.1 Patch 4 supercede all previous general patches for WebObjects. If you have already installed a previous WebObjects patch on your system, you can install Patch 4 over it. If you have not installed any previous patch, you do not need to; this patch contains all fixes from previous patches.

If you use Enterprise Objects Framework on your system, it should be installed before you install this patch. This patch includes fixes in system components that were originally installed by EOF.

Please read the appropriate section below for details on downloading and installing the patch.

Installation instructions - OPENSTEP for Mach

WebObjects users on OPENSTEP 4.2 for Mach must install OPENSTEP 4.2 for Mach Patch 4 on their system to avoid Year 2000 problems. If you received a WebObjects Patch 4 CD, this patch and its documentation should be included on the CD. Otherwise, you can download the patch from Apple's web site. For details visit:

http://til.info.apple.com/techinfo.nsf/artnum/n70033

Mach users do not need to download or install any additional binary patches for WebObjects. However, WebObjects developers on Mach should read sections 4 and 5 of this document carefully. There are manual steps you must take to avoid some century issues.

Installation instructions - Windows NT 4.0

The patch installer for WebObjects 3.5.1 Patch 4 for Windows NT is named WO351Patch4Installer.exe (CD version) or WO_351_Patch4.exe (compressed web version). If you download the patch from the internet, you must then run the executable to uncompress the patch installer, WO351Patch4Installer.exe, and release notes. Be sure to read the release notes carefully before installing this patch.

To install this patch on a Windows NT 4 system, perform the following steps:

1. Log in as a user with Administrator privileges.
2. Make sure that there are no WebObjects, OPENSTEP or EOF applications running.
3. Double-click on the patch installer, WO351Patch4Installer.exe, to start the install process.
4. A screen will appear with some information about this patch. After you have read it, click “Next”to continue.
5. The license agreement for this patch will appear. To agree to the license and continue the installation, click “Yes”.
6. The patch will now be installed on your system and you will be asked if you want to reboot. Select “Yes, I want to restart my computer now” to reboot your Windows NT system. The WinZip self-extractor will automatically quit and remove the temporary installation in about 20 seconds.
7. For WebObjects developer installations, review sections 4 and 5 of this document. There are manual steps you must take to avoid some century issues.

Installation instructions - OPENSTEP Enterprise for Solaris and HP-UX

The patch installer for WebObjects 3.5.1 Patch 4 for Solaris is named WO351SolarisPatch4.tar.Z. For HP-UX, the patch is named WO351HPUXPatch4.tar.Z. Be sure to read the release notes carefully before installing this patch.

To install this patch on a Solaris or HP-UX system, perform the following steps:

1. Log in as root. If you’re not sure how to log in as root, see your system administrator.

2. Make sure that there are no WebObjects, OPENSTEP or EOF applications running.

3. Change to the directory containing the patch and the patch installer, “patcher.sh”.

4. At a shell prompt, type:

   ```
   patcher.sh –install WO351SolarisPatch4.TAR.Z
   ```

   or

   ```
   patcher.sh –install WO351HPUXPatch4.TAR.Z
   ```

   For more information on using the “patcher.sh” program, type the following at the command prompt:

   ```
   patcher.sh –help
   ```

5. Reboot your Unix system.

Section 2: Issues - WebObjects 3.5 Patch 4

This section details all issues addressed in WebObjects 3.5.1 Patch 4 and in previous patches for WebObjects. Please see section 4 as well for details on some manual steps you may need to take to avoid some Year 2000 problems.

Issues addressed in WebObjects 3.5 Patch 4

"tar" and "gnutar" commands didn't accept dates later than 1999
Apple reference #2342167, 2342424, 2342425

Problem: The "gnutar" and "tar" commands have an option to only write files newer than a given date. The versions of "gnutar" and "tar" provided with WebObjects 3.1 didn't accept dates past the end of 1999, nor did they accept a four-digit year format for the date argument.

Resolution: The "gnutar" and "tar" commands have been changed to correctly handle two- and four-digit year arguments to the -N option, and to process all dates in the valid UNIX range (1970-2068).

No Euro glyph support in OPENSTEP rulebooks
Apple reference #2265773

Problem: The AppKit rulebooks did not support the Euro currency character.

Resolution: The rulebooks have been updated to provide support for the Euro currency symbol, U+20AC. The encoding tables in Foundation have also been updated to support this character. You will still need to obtain and install fonts which include the Euro character in order to generate the character itself.

NSDateFormatter interpreted year “00” as “Current Year”

Apple reference #2278598

Problem: NSCalendarDate interpreted year “0” or “00” as “current year”, rather than “2000”, which was inconsistent with the interpretation of two-digit years in UNIX and not Year 2000 compliant.

Resolution: NSCalendarDate now interprets “0” or “00” year values as “2000”.

Date formatter handled two digit year inappropriately

Apple reference #2290369

Problem: NSCalendarDate treated any two-digit year as a 20th century year.

Resolution: NSCalendarDate now treats two-digit year values less than 30 as 21st century years.

Euro glyph didnít display correctly

Apple reference #2298782

Problem: NSTextView’s -insertText: method handled mutable attributed strings incorrectly under WebObjects. This resulted in display problems for certain characters, including the Euro currency symbol.

Resolution: This method now handles attributed strings as well as ordinary strings correctly.

Issues addressed in WebObjects 3.5.1 Patch 3

Some exceptions not passed across Java Bridge

Apple reference # 2300566

Problem: If an Objective C exception was raised in an Objective C method called from Java, the bridge correctly converted the NSException to a Java exception and threw it in Java. If the conversion of the NSException itself raised an exception, the subsequent behaviour of the application was undefined.

Resolution: The conversion exception is now caught, both exceptions are logged, and a Java UnknownError is thrown in the Java VM with the message: “Failed to convert Objective C exception to Java exception. Refer to log output for more information”.

Java access violation error when bridge cannot find a class

Apple reference # 2303014

Problem: If a Java exception occurred while WebObjects was attempting to look up a Java class, the exception might not be re-raised in Objective C. If this occured, the JavaBridge sometimes crashed when it failed to load the wrapper class.

Resolution: The JavaBridge now raises an appropriate exception in Objective C when a wrapper class can not be found.

Java object creation race condition

Apple reference # 2310235

Problem: A race condition on Windows NT sometimes allowed the Java garbage collector to reclaim a Java object while it was still needed by Objective C code, resulting in application crashes.

Resolution: The integration between Java and Objective C during the creation of hybrid objects has been improved to decrease the likelihood of premature garbage collection.

Garbage collector frees wrapped objects prematurely

Apple reference # 2310270

Problem: When a wrapped Objective C object was created in a Java method, the Objective C proxy object was not retained. Java garbage collection could then release this object while it was still in scope.

Resolution: Objective C proxy objects are now retained until the appropriate autorelease pool is deallocated.

next.util.CalendarDate constructors calculate year values incorrectly

Apple reference # 2310340

Problem: Under WebObjects 3.5.1, next.util.CalendarDate constructors did not correctly handle the two-digit year values returned by by the Java Date class' getYear() method, causing CalendarDate objects to be constructed with incorrect year values..

Resolution: These next.util.CalendarDate constructors now calculate year values correctly.

Java Bridge fails when an Objective C constructor returns nil

Apple reference # 2310395

Problem: The Java Bridge did not correctly handle the case where the Objective C base of a hybrid object returned nil from its Objective C init method.

Resolution: In this case, a NullPointerException is now raised in Java. This fix included a change to one of the header files used by the "bridget" utility. All Java projects must be rebuilt to take advantage of this fix. Please note that your code still needs to wrap Java constructors in try/catch statements to catch instantiation exceptions; see TIL article 70035 for more details.

NSUnicodeStrings not correctly converted by Java Bridge

Apple reference # 2310656

Problem: In WebObjects 3.5.1, NSUnicodeStrings were not correctly converted into Java strings.

Resolution: Unicode strings are now detected and correctly converted into Java strings.

AutoreleasePool.finalize() releases Objective C autorelease pool

Apple reference # 2310883

Problem: The next.util.AutoreleasePool.finalize() method contained code that released the underlying Objective C autorelease pool. Although the intention of this code was to prevent leaks, it could result in an Objective C autorelease pool being released unexpectedly by routine Java garbage collection if a reference to the Java pool was not held.

Resolution: The Objective C autorelease pool is no longer released in AutoreleasePool.finalize(). If AutoreleasePool.release() is not explicitly called, the underlying Objective C pool will leak into its containing pool. Refer to TIL article 70035 for more information about proper use of autorelease pools in Java.

Issues addressed in WebObjects 3.5.1 Java Patch 2

DecimalNumber

Several of the next.util.DecimalNumber arithmetic methods raised exceptions when called. These methods now work properly.

CalendarDate

The next.util.CalendarDate constructor CalendarDate(java.util.Date date) would sometimes return an incorrect date. This has been fixed.

NoSuchMethodError exceptions

WebObjects applications written in Java would occasionally raise a java.lang.NoSuchMethodError exception, when the method did in fact exist. This was caused by a race condition on the interaction between Objective C's reference counting and Java's garbage collection. This has been fixed. These exceptions could also be raised if a subclass of next.eo.CustomObject did not implement a default constructor (that is, an -init method with no parameters). Chained initializer methods would give the same result if some of the methods in the chain did not have Java equivalents. After installing this patch, a missing default constructor will generate a more meaningful error message.

Java string retain crashes

A problem was fixed where Java strings were garbage collected before WebObjects had a chance to retain them. The missing strings caused some applications to crash under heavy use.

Multiprocessor crashes on Solaris

Because of improper use of lock files, WebObjects 3.5.1 applications would sometimes crash on multiprocessor Solaris machines. This behavior is fixed in this patch.

Incorrect cloning behavior

Previously, custom Java objects did not correctly implement the Cloneable interface. With this patch installed, subclasses of NextObject and CustomObject should be cloneable.

DirectToWeb application crashing

The previous version of this patch for Windows NT caused crashes when launching a DirectToWeb application. This has been corrected in the current patch release.

EOEditingContext can now retain all referenced objects

EOEditingContext's weak references to objects caused problems with Java's multithreaded garbage collection. If you are writing an application which accesses the Enterprise Objects Framework from Java, you may encounter two problems:

- EOEditingContext's processing of an EO's deallocation notification is not thread safe
- An object that is no longer retained by your application but has not been deallocated may be refetched. However, the reference may be reestablished too late to prevent the object from being deallocated.

In order to solve this problem, we have made it possible for EOEditingContext to retain all of its objects. Two class methods have been added:

```
+ (void)setInstancesRetainRegisteredObjects:(BOOL)flag;

+ (BOOL)instancesRetainRegisteredObjects;
```

If instancesRetainRegisteredObjects is YES, newly created EOEditingContexts will retain any EOs that it is observing. The default value for this flag is YES if an application contains Java, and NO otherwise. Each EOEditingContext caches the value of this flag, so it is possible to switch from one state to the other midway through your application. Note that only newly created EOEditingContexts will be affected by the flag.

In order to collect the EOs, you must either deallocate the editing context, or call the new method:

```
- (void)reset;
```

on the editing context. Calling reset causes the editing context to forget and release all of the objects that it is watching, and makes all of those objects effectively unusable. You typically will not not have to do anything to a Java WebObjects application in order to be compatible with this new behavior. A session's defaultEditingContext will get collected when the session times out, releasing all of the EOs fetched by that session.

Issues addressed in other WebObjects patches

Missing header files in WebObjects 3.1

Apple reference # (none)

Problem: Certain header files, required to subclass the WOAdaptor and WODefaultAdaptor classes, were missing from WebObjects 3.1.

Resolution: These header files are installed by the current patch.

Section 3: Issues - WebObjects Examples

In addition to the problems addressed in WebObjects 3.1 and 3.5.1 Patch 4, some examples shipped with WebObjects 3.1 and 3.5.1 Developer use time and date objects in ways which are not consistent with Apple's recommended coding practice. This section contains instructions on how to change the WebObjects examples to avoid this problem.

CyberWind_J example uses java.util.Date API in non-Y2K-compliant manner
Apple reference #2348777, 2350204

Problem: The Japanese version of the WebObjects Cyberwind example displays dates with a two-digit year format in two places. This is non-preferred coding practice.

Resolution: Customers who need for this example to be Y2K compliant can achieve this by making the following changes in the example source and recompiling.

Change "%y" in the date format string to "%Y" in the following locations:

{WODOCUMENTROOT}/WebObjects/Examples/WebScript/CyberWind_J/CyberWind_J.woaApplication.wos, line 57

{WODOCUMENTROOT}/WebObjects/Examples/WebScript/CyberWind_J/Footer.wo/Footer.wos, line 23

WOJavaExamples use java.util.Date API in non-Y2K-compliant manner
Apple reference #2348779

Problem: The WebObjects Java examples "ComponentEditorJava" and "FdfJava" call java.util.Date.getYear() in several places, without correcting for the fact that the value returned by this method is "years since 1900". The result is two-digit years through 1999, and three-digit years thereafter, which is not Y2K compliant.

Resolution: Customers who need for these two examples to be Y2K compliant may achieve this by making the following modifications in the example source and recompiling.

Add 1900 to the value returned by java.util.Date.getYear() in the following locations:

{WODOCUMENTROOT}/WebObjects/Examples/Java/ComponentEditorJava/CommonJava/ComponentNode.java, line 59

Change:

```
return name()+"|"+d.getMonth()+"/"+d.getDate()+"/
"+d.getYear()+"|"+file().length();
```

To:

```
return name()+"|"+d.getMonth()+"/"+d.getDate()+"/
"+(d.getYear()+1900)+"|"+file().length();
```

{WODOCUMENTROOT}/WebObjects/Examples/Java/FdfJava/TenFortyEZ.java, line 115

Change:

```
return " "+d.getMonth()+"/"+d.getDate()+"/
"+d.getYear();
```

To:

```
return " "+d.getMonth()+"/"+d.getDate()+"/
"+(d.getYear()+1900);
```

WebObjects' DynamicElements example uses java.util.Date API in non-Y2K-compliant manner

Apple reference #2384444

Problem: The WebObjects DynamicElements example calls java.util.Date.getYear() in one place without correcting for the fact that the value returned by this method is "years since 1900". The result is two-digit years through 1999, and three-digit years thereafter, which is not Y2K compliant.

Resolution: Customers who need for this example to be Y2K compliant may achieve this by making the following modifications in the example source and recompiling.

Add 1900 to the value returned by java.util.Date.getYear() in the following location:

{WODOCUMENTROOT}/WebObjects/Examples/WebScript/DynamicElements.woa/JavaScriptEx1.wo/JavaScriptEx1.wos

Change:

```
document.write(\"Calculated on: \",
today.getMonth()+1,\"/\",today.getDate(),\"/\",
today.getYear());
```

To:

```
document.write(\"Calculated on: \",
today.getMonth()+1,\"/\",today.getDate(),\"/\",
today.getYear()+1900);
```

Example data contains two-digit year values
Apple reference 2384449

Problem: The WebObjects EmployeBook and EmployeeBookJava examples' Department contains date values with two-digit years, which is not preferred practice.

Customers can edit the Department data to use a four-digit year format. The two files which need to be modified are:

{WODOCUMENTROOT}/WebObjects/Examples/Java/EmployeeBookJava/Departments.array

{WODOCUMENTROOT}/WebObjects/Examples/WebScript/EmployeeBook/Departments.array

For example, the following data:

```
{ name = "Administration"; location = "Hillside 1"; employees = (
          { employeeID = 126; departmentName = "Administration";
departmentLocation = "Hillside 1"; lastName = "Seng"; firstName
= "Paul"; eMail = "Paul_Seng@NTeX.COM"; phoneNumber = "415-320-
7249"; address = "4876 High Road"; city = "Half Moon Bay"; state
= "CA"; zip = "94171"; title = "Designer"; hireDate = "02-JUL-
88"; salary
= 5000.00; },
```

should be changed to:

```
{ name = "Administration"; location = "Hillside 1"; employees = (
            { employeeID = 126; departmentName =
"Administration"; departmentLocation = "Hillside 1"; lastName =
"Seng"; firstName = "Paul"; eMail = "Paul_Seng@NTeX.COM";
phoneNumber = "415-320-7249"; address = "4876 High Road"; city =
"Half Moon Bay"; state = "CA"; zip = "94171"; title = "Designer";
hireDate = "02-JUL-1988"; salary
= 5000.00; },
```

The data may be modified manually using any text editor, or by the use of "sed", as demonstrated below:

```
my Departments.array Departments.array.bak

sed 's/[0-9][0-9]\-[A-Z][A-Z][A-Z]\-/&19/g'
Departments.array.bak > Departments.array
```

Section 4: Issues - Enterprise Objects Framework

This section discusses century issues in Apple's Enterprise Objects Framework.

Apple is not aware of any century problems with Enterprise Objects Framework versions 1.1, 1.2, 2.1, or 2.2 which would affect development or deployment of EOF applications. However, some examples shipped with these versions of EOF use two-digit year formatting, which is not consistent with Apple's recommended coding practice. Instructions on how to change the EOF examples to avoid this problem can be found in Apple's Tech Info Library. For EOF 1.1 and 1.2, please see TIL article 70084, at:

http://til.info.apple.com/techinfo.nsf/artnum/n70084

For EOF 2.1 and 2.2, please see TIL article 70085, at:

http://til.info.apple.com/techinfo.nsf/artnum/n70085

OPENSTEP and EOF share many components; in order to ensure that EOF is Year 2000 compliant, you must install the appropriate OPENSTEP or OPENSTEP Enterprise patch on your system, following the install order given in section 1 of this document.