WebObjects Framework Release Notes

For Release 3.5
Last Updated 11/26/97

This file contains notes about the WebObjects family of products. WebObjects release 3.5 is the current release for the Window NT, Solaris, HP-UX, and Mach platforms.

Also refer to these other release notes:

- WebObjects development tools release notes (known problems with Project Builder, Direct to Web, and WebObjects Builder)
- Enterprise Objects release notes (known problems with the Enterprise Objects Framework and the EOModeler application).

Server Compatibility

WebObjects has been tested in these configurations. In addition, WebObjects should work with other servers, provided they follow the CGI or NSAPI specifications. The numbers in parentheses are references to bugs listed in the next section.

·€Netscape Enterprise + CGI adaptor (Windows NT Workstation or Server and UNIX platforms)

·€Netscape Enterprise + NSAPI adaptor (Windows NT Workstation or Server and UNIX platforms)

·€Netscape Fasttrack + CGI adaptor (UNIX platforms)

·€Netscape Fasttrack + NSAPI adaptor(UNIX platforms)

·€Microsoft IIS Server + CGI adaptor (Windows NT Workstation or Server)

·€Microsoft IIS Server + ISAPI adaptor (Windows NT Workstation or Server) (installed as 65135 and 78035)

·€PWS Server + CGI adaptor (Windows NT Workstation or Server)

·€PWS Server + ISAPI adaptor   (Windows NT Workstation or Server) (installed as 65135 and 78035)

·€Apache + CGI adaptor   (Windows NT Workstation or Server)

Known Bugs

This section lists bugs that we are aware of with this release and suggests ways to avoid or work around these problems. Please refer to the bug reference number if you need to contact Apple about a bug's status.

Installation Known Problems

Reference: 82237

Problem: WebObjects 3.5 installation overwrites examples

Description: If you have modified any WebObjects examples from a previous installation of WebObjects, they will be removed, along with the unmodified WebObjects examples, from the document root of your web server when you install WebObjects 3.5.

Workaround: Move any examples that you'd like to preserve to a different location before uninstalling or reinstalling WebObjects.

Reference: 82124

Problem: On Windows NT, the WebObjects Deployment installation does not add *NEXT_ROOT***\NextLibrary\JDK\bin** to PATH

Description: WebObjects applications containing server-side Java code require the directory *NEXT_ROOT***\NextLibrary\JDK\bin** in the PATH environment variable.

Workaround: Modify the PATH environment variable to include the following at the end of the path list:

*NEXT_ROOT***\NextLibrary\JDK\bin**

where *NEXT_ROOT* with the location of your WebObjects installation.   For example, if you installed WebObjects in the default location, **C:\NeXT**, you'd add the following to the end of the PATH:

C:\NeXT\NextLibrary\JDK\bin

Reference: 65135

Problem: Cannot autostart or load-balance WebObjects applications with Microsoft Internet Information Server (IIS) or Peer Web Server (PWS)

Description: This note applies to the Microsoft IIS server on NT 4.0 Server and the PWS server on NT 4.0 Workstation.

If you install WebObjects in a directory other than the root directory, the CGI adaptor won't autostart your applications correctly. The server creates subprocesses with a special user that has no privileges and that has no access to the **NEXT_ROOT** environment variable. An autostarted application won't have any privileges either and won't be able to locate OpenStep resource files (for example, time zone files).

If you don't use the following workaround, you will not be able to autostart applications using the CGI adaptor and you won't be able to use the ISAPI adaptor for load balancing.

Another unrelated but interesting issue is that sometimes the server does not recognize the WebObjects CGI adaptor unless you specify the **.exe** extension. This means that an URL to start a WebObjects application using the CGI adaptor would have to be:

http://localhost/Scripts/WebObjects.exe/*MyApp*

Workaround: Do the following:

1) Install WebObjects software under the root directory of one of your machine's local hard drives. When the installer asks you for a directory in which to install WebObjects software, click the "Browse..." button and use the resulting dialog box to select the top-level directory (e.g. **C:\**) for one of your machine's drives.

2) Give administrator privileges to the default CGI user set up by the Microsoft web server installer. Using the "User Manager for Domains" application under Administrative Tools, double-click the entry with the full name "Internet Guest Account." (The username usually starts with "IUSR_".) In the window that opens, click the Groups button. You'll see two tables, one called "Member of" and one called "Not member of." Move the Administrators group from the "Not member of" table to the "Member of" table. Click the OK button.

Reference: 73185

Problem: Windows NT Installer fails if cgi-bin or document root contain spaces

Description: When installing WebObjects on Windows NT 4.0, installation will fail if your cgi-bin directory or document root directory has a path containing any whitespace.

Workaround: Change your paths so that they do not contain any whitespace.

Reference: 78950

Problem: Installing from the same disk requires extra free space

Description: On Windows NT, if you copy the CD image onto your local disk drive and run the installer from there, you will need roughly three times the free space required by the release.

Workaround: Always install from the CD, or install from one local disk onto another local disk.

Reference: 82082

Problem: Free space required on system drive in addition to installation drive during install

Description: On Windows NT, if you install WebObjects on a drive that is different from your system drive (that is, the one that contains the **WINNT** directory), you still need to have free space on your system drive or the install will fail.

Workaround: The install process creates a directory tree named **WebObjects** on your system drive. This directory contains the Java class files. You can delete this directory tree, as it has also been installed on the disk that you chose during the installation. If the installation fails, delete unnecessary files from your system drive and then try again.

Reference: 72353

Problem: Cannot use forward slashes when specifying path names in Windows NT installer.

Description: When installing WebObjects on Windows NT, you are prompted for your server's cgi-bin and document root paths. If you choose to type in the path without going through the file system browser, you cannot type the path with forward slashes, as in **C:/cgipath**. If you do, the paths will become corrupted in the registry.

Workaround: Use backslashes when typing paths in the installer, as in **C:\cgipath**.

Reference: 78035

Problem: On Windows NT, **machd** and **nmserver** are not installed properly

Description: WebObjects applications, WebObjects Builder, and EOModeler require the NT services **machd** and **nmserver** be running. WebObjects installs **machd** and **nmserver** but fails to install them as NT services. As a result, when you try to start up an application, it won't run.

Workaround: Open a Bourne shell window or a DOS Command prompt window. Enter the following commands:

```
> cd %NEXT_ROOT%\NextLibrary\System
> machd.exe -install
> nmserver.exe -install
```

Reference: 82275

Problem: WebObjects Support bundle no longer needed

Description: Previous releases of WebObjects required you to load a bundle into Project Builder named **WebObjectsSupport.bundle**. This bundle is no longer necessary in WebObjects 3.5. If you uninstall the previous release, **WebObjectsSupport.bundle** is deleted. After you install WebObjects 3.5, you may receive an error message the first time you launch Project Builder saying that it cannot find **WebObjectsSupport.bundle**.

Workaround: Go to the Project Builder Preferences panel. Choose Bundles from the pop-up list. Remove **WebObjectsSupport.bundle** from this list.

Reference: 73668

Problem: PDO Apache installation conflicts with PDO WebObjects installation.

Description: The Apache installation script wants to install Apache to **/usr/NextLibrary**. WebObjects doesn't install things in **/usr** anymore, and it creates a symbolic link in **/usr** to the real location of **NextLibrary**.

Workaround: Install the Apache server after installing WebObjects. Note that you must know what the cgi-bin and document root will be ahead of time, as the WebObjects installation script prompts you for this information.

Reference: 82177

Problem: On Solaris and HP-UX, **RebuildWODefaultApp** assumes 7.3 static linking for Oracle adaptor

Description: The **RebuildWODefaultApp** script contains no setting for the ORACLE_REL makefile variable. The default value for ORACLE_REL is 7.3-static. This causes the rebuild of **WODefaultApp** to fail if you don't have those client libraries installed.

Workaround: A version of **RebuildWODefaultApp** with a fix for this bug is available through NeXTanswers #2555. Retrieve the script from NeXTanswers and run it by hand after installing WebObjects 3.5.

Alternatively, you can set an environment variable named ORACLE_REL to one of the following values:

- "8.0-static", for Oracle 8.0.3 static client library linking
- "7.3-dynamic", for Oracle 7.3.2 dynamic client library linking

Reference: 81923

Problem: To use **make** on Solaris and HP-UX, you must have **/usr/local/bin** in your path

Description: The default path for the root user on Solaris and HP-UX doesn't necessarily include **/usr/local/bin**. To use the appropriate **make** facility for WebObjects applications, you must have **/usr/local/bin** in your path ahead of any directory that might include an executable or link to an executable called **make**.

Workaround: Check the path of the root user. If it doesn't include **/usr/local/bin**, type **which make**. Add **/usr/local/bin** to the path ahead of whatever **which make** returns.

Reference: 81829

Problem: The WebObjects examples are not distributed built on HP-UX.

Description: The WebObjects examples are not distributed in built form on HP-UX as they would take up too much disk space.

Workaround: Build any examples that you want to run by hand.

Reference: 79798

Problem: Uninstall on Windows NT deletes parts of Project Builder, EOModeler, and ProjectBuilder frameworks

Description: If you've installed WebObjects on top of OpenStep Enterprise, be aware that uninstalling WebObjects will remove certain parts of OpenStep, including key parts of Project Builder and EOModeler. WebObjects installs newer versions of this software and overwrites the older versions.

Workaround: Reinstall WebObjects or OpenStep Enterprise after uninstalling WebObjects.

Reference: 74365

Problem: Deployment installer allows user to enter Developer serial numbers

Description: It is possible to enter your WebObjects Developer serial number when doing installs from a WebObjects Deployment CD. Much of the Developer software (such as Developer applications like WebObjects Builder) will be missing from what looks like a successful, typical WebObjects Developer installation.

Workaround: If you've installed what you think is WebObjects Developer but you're missing crucial parts of the software (such as WebObjects Builder), ensure that the CD is titled "Developer" and not "Deployment".

Reference: 72264

Problem: On Windows NT, Uninstaller fails to remove much of WebObjects

Description: When you run the WebObjects Uninstaller on Windows NT, often much of the WebObjects release remains on your system. In particular, items are not removed from the Windows NT Registry .

Workaround: A "Cleanup Kit" is available through NeXTanswers #2556. Alternatively, you can follow the instructions for uninstalling WebObjects on Windows NT in the Post-Install Instructions, available online.

WebObjects Adaptors Known Problems

Reference: 72343

Problem: Application is inaccessible if incorrect application side adaptor is specified

Description: By default, WebObjects applications use WODefaultAdaptor as the adaptor class. If you have created your own adaptor class, you can specify it on the command line using the **-a** option. If the application executable cannot find the specified adaptor class at runtime, the WebObjects application is unreachable. This means if you don't capitalize the class name properly or if you misspell it, your application will be inaccessible and will provide no reason for failure.

Workaround: Check the name you specified with the **-a** option and verify that it is correct.

Reference: 77878

Problem: **WebObjects.conf** cannot contain textual comments.

Description: **WebObjects.conf** only allows you to comment out statements that would otherwise be syntactically correct. For example, the following is an acceptable comment because if you removed the #, the line would be syntactically correct.

```
# Examples/HelloWorld:1@faux.apple.com 3000
```

You cannot have a comment meant to be read by humans. For example, this type of comment is not allowed:

```
# faux.apple.com is out of service. This is a BAD comment line.
```

Workaround: None

Reference: 74404

Problem: WebObjects log file is not updated.

Description: This bug occurs for NSAPI adaptors on all platforms and the ISAPI adaptor on Windows NT.
For performance reasons, when the adaptor receives its first log function call, it sets a global "attempted logging flag" to YES and checks to see if there is a **logWebObjects** file in the temporary directory (usually **c:\temp** on Windows NT and **/tmp** on other platforms). If the file exists, it sets a global "log flag" to YES. If the file doesn't exist, it sets the "log flag" to NO. On subsequent log function calls, the adaptor will only log a message if both the "attempted logging flag" and the "log flag" are YES.

The problem arises from the fact that these flags are global variables. In threaded API servers, if one thread has set the "log flag" to NO, there is no way it will be turned back to YES.

Workaround: Do the following:

- Create the **logWebObjects** file in the temporary directory.

- Restart your server.
- If you remove the **logWebObjects** file, you should restart your server afterward.

This workaround will work as long as the server keeps using the current thread for the API adaptor.

Reference: 82214

Problem: NSAPI 3.0 adaptor improperly linked on HP-UX

Description: On HP-UX when setting up the NSAPI 3.0 adaptor for use with Netscape Enterprise Server 3.01, after you modify the **obj.conf** file and restart the server, you receive errors complaining about unresolved symbols in the NSAPI adaptor.

Workaround: Rebuild the NSAPI 3.0 adaptor from the source code (in **/NextLibrary/WOAdaptors/NSAPI/3.0/Source**), replace the existing NSAPI adaptor with the new one, and restart the server.

Reference: 82245

Problem: Recording crashes on HP-UX

Description: The **WORecording** adaptor crashes on HP-UX.

Workaround: Record on any other platform than HP-UX, and then playback on HP-UX when needed.

Reference: 82035

Problem: Makefile for WOPlayback erroneously refers to **gcc** instead of **cc**

Description: The source code for the WOPlayback tool is included in the WebObjects 3.5 release. If you make modifications to WOPlayback and try to build it, you receive an error message:

```
Make: Cannot load gcc.  Stop.
*** Exit 1
Stop.
```

Workaround: Replace **gcc** with **cc** in WOPlayback's Makefile.

Database Access Known Problems

Reference: 73644

Problem: Can't backtrack in applications that access a database.

Description: If you try to backtrack using the browser in a database application, it doesn't work because the WODisplayGroup object is out of sync with the display.

Workaround: Disable browser backtracking using WOApplication's **setPageRefreshOnBacktrackingEnabled:** method.

Reference: 75766

Problem: Autostarted applications using ODBC can't contact the database.

Description: Frequently, the ODBC data sources are configured as "user" data sources, which are available only to the user that created them. Autostarted WebObjects applications run with the permissions and environment of the "anonymous" user specified by the web server used to start the application. If this anonymous user is not the same as the user who created the user data source, the application will not be able to use the data source.

Workaround: The best solution is to configure a "system" data source, which would be available to any process in your system. To do so:

1. Install the most current ODBC drivers and related support software available to you.
2. Open the ODBC Control Panel.

3. Click the "System DSN" tab, and click the "Add" button to add a new data source.

If your ODBC support software doesn't have this feature and you are unable to acquire a newer version of ODBC, log in as the "anonymous" user (you may have to do some configuration to get this to work) and configure a user data source that points to your database.

Reference: 81473

Problem: On Solaris and HP-UX, the EOAdaptor makefile does not handle Oracle 8.0 by default.

Description: By default, the static linking makefiles used on Solaris and HP-UX uses Oracle 7.3 static libraries. To support Oracle 8.0, you must set the macro ORACLE_REL in your **Makefile.postamble** to be either:

```
ORACLE_REL = 7.3-dynamic
ORACLE_REL = 8.0-static
```

If ORACLE_REL is undefined, it defaults to 7.3-static.

Workaround: None

Reference: 82130

Problem: Accessing display groups queries using key-value paths always retrieves all the objects

Description: If you use a key-value path starting with a WODisplayGroup (such as displayGroup.queryMatch.xx.yy) the query always retrieves all of the objects.

Workaround: Implement accessor methods in your component's code and use those methods to access the query values. For example, if you have **displayGroup.queryMatch.department.name** in your component's **.wod** file, you would replace it with:

```
MyField: WOTextField {
      value=displayGroupQueryMatchDepartmentName
}
```

and implement the following two methods in your component's code:

```
public Object displayGroupQueryMatchDepartmentName()
{
      return displayGroup.queryMatch().valueForKey("department.name");
}

public void setDisplayGroupQueryMatchDepartmentName(Object value)
{
      displayGroup.queryMatch().takeValueForKey(value, "department.name");
}
```

Monitor Application Known Problems

Entry: 82252

Problem: Problems with Monitor on HP-UX

Description: On HP-UX, if you attempt to launch an application using Monitor and MonitorProxy and the application fails to launch, Monitor won't allow you to change the settings for that application.

Workaround: Terminate both Monitor and MonitorProxy and restart them. Make your changes to the application settings before attempting to launch.

Reference: 80534

Problem: Manually editing the public configuration file changes Monitor behavior

Description: If you manually remove items from the public configuration file **WebObjects.conf** or you delete the **WebObjects.conf** file altogether, the Monitor will remove these items from its interface. This behavior is required for a feature in Monitor to work. When Monitor launches, it automatically starts instances that are set to auto-recover if it finds that they are not running. Therefore Monitor will check the **WebObjects.conf** file at startup time and merge what it finds with its internal preferences.

Workaround: None. As a rule, you should not try to edit the **WebObjects.conf,** as Monitor is really a manager of this file.

Reference: 81677

Problem: Newly added instances do not contain the proper Application Path

Description: When you add another instance to an application in Monitor, Monitor does not use the path settings from the first instance when it creates the new instance. This is because Monitor doesn't know if the application is intended to run on the same settings as the first instance; it may need to run on a different server that has a different operating system.

Workaround: Click the "More" button for that instance to display the Application Instance inspector. Type the application path in the Application Path field. You can copy and paste the value from the first instance's Application Path field.

Reference: 82094

Problem: Deleting an instance in Monitor does not terminate that instance

Description: If you click the Delete button in Monitor to remove an application, it does not terminate any instances of that application that are running.

Workaround: None

Reference: 81916

Problem: When using Monitor on IIS, links to running applications don't work

Description: Links to running applications from the Monitor don't work if you're using the IIS server. The links to running applications are of the form"http://hostname/cgi-bin/WebObjects/..." . This is not the correct URL for IIS.

Workaround: Go to "Options" in Monitor and changed the adaptor key from "cgi-bin/WebObjects" to "scripts/WebObjects.exe".

Reference: 81276

Problem: When scheduling is on, applications abruptly terminate instead of gracefully shutting down

Description: When scheduling is used on an application instance in the Monitor application, the instance is stopped using the method **terminateFromMonitor** rather than by setting **refuseNewSessions:YES** and terminating when sessions have timed out. Thus, any sessions active when Monitor has scheduled the application to shut down are lost.

Workaround: Don't use Monitor for scheduling if you want applications to shut down gracefully. New API in WOApplication and other classes allow the application to determine when to shut down. The following example of a **sleep** method implemented in your Application class provides a way to have an application begin to refuse new sessions after 6 hours of running time. After all the sessions have timed out, the application will terminate itself:

```
- sleep
{
    [super sleep];
    id stats = [self statistics];
    NSDate *appStartDate = [stats objectForKey:@"StartedAt"];
    NSTimeInterval timeSinceStarted = ([appStartDate timeIntervalSinceNow] * -1); //in seconds
    if (timeSinceStarted > 21600) {     // 21600 sec = 6 hours
        [self setMinimumActiveSessionsCount:0];
        [self refuseNewSessions:YES];
    }
}
```

For more information, see the "Deployment and Performance Issues" chapter of the *WebObjects Developer's Guide*.

Reference: 82066

Problem: Refuse New Sessions should be inactive when instance is not running

Description: Monitor allows you to click the Refuse New Sessions button for application instances that aren't currently running. If you do so and then start the instance, it won't accept a session. If a new session attempts to contact the application, the instance terminates.

Workaround: Only click Refuse New Sessions for running instances.

Reference: 82010

Problem: Monitor shuts down applications differently depending on the platform

Description: If you shut down a running instance in Monitor by clicking the On/Off toggle button, the Monitor application sends a Distributed Objects message to the running instance. This method has different behavior on NT and Mach than on Solaris and HP-UX. On NT and Mach it invokes the WOApplication **terminate** method. On Solaris and HP-UX it calls **exit(0)**.

Workaround: None

Miscellaneous Known Problems

Reference: 82291

Problem: **gdb** breaks on unknown signal on Solaris and HP-UX

Description: **gdb** breaks at the end of every request-response loop cycle, claiming that it has caught an unknown signal in **closesocket**.

Workaround: A **gdb.ini** file that fixes this problem is created in the directory of every WebObjectsApplication project. To ensure that this file is loaded into **gdb**, always run **gdb** from the project directory. For example:

```
% cd projectName
% gdb
(gdb) run projectName.woa/projectName
```

Reference: 78562

Problem: Mach 4.1 and 4.2: high load caused kernel panic

Description: Under heavy load situations, starting at approximately three hits per second aimed at a single WebObjects server on Mach 4.1 or 4.2, a CPU(0):InternalEntryAllocate panic consistently occurs.

Workaround: No known workaround at this time.

Reference: 80689

Problem: Application launch mechanism doesn't handle symbolic links under document root

Description: You can launch most WebObjects applications by specifying their executable names with no arguments on the command line. However, if the application is symbolically linked into the document root, instead of physically under the document root, you must supply an argument so that the application name can be determined correctly. Symbolic links resolve to a path that doesn't appear to be under the document root. For example, all example applications on Solaris or HP-UX will have this problem because the examples directory is a symbolic link to **/NextLibrary/WODocumentRoot**. Your application may appear to operate normally, but images won't load because they won't be found.

Workaround: Specify the application name on the command line when launching. Consider the HelloWorld example:

```
% cd /NextLibrary/WebServer/htdocs/WebObjects/Examples/WebScript/HelloWorld/HelloWorld.woa
% HelloWorld Examples/WebScript/HelloWorld
```

Reference: 74772

Problem: WebObjects treats text inside <script></script> tags as HTML

Description: When dynamically generating HTML, WebObjects tries to display the text inside of a <script> tag as HTML. If you use a "<" or ">" operator in the script, WebObjects converts the operator to &lt; or &gt;, respectively, causing the script to fail on the client.

Workaround: To include script in a component, use the WOJavaScript or WOVBScript dynamic element.

Reference: 80879

Problem: Unbalanced WEBOBJECT tags are ignored and sometimes cause strange behavior

Description: Typos in your HTML involving the WEBOBJECT opening and closing tags can cause WebObjects to ignore parts of your HTML.

Workaround: Be sure to balance your tags.

Reference: 76296

Problem: WOF has APIs that are incompatible with the C++ compiler

Description: For instance the word "template" is reserved in C++. When such a conflict occurs, the C++ compiler stalls.

Workaround: Comment the faulty line in the header, and if you need to reference this API, create a copy of the header (one that has the API one that does not).

Reference: 81730

Problem: WOStats Memory statistics are broken on HP-UX

Description: There is no memory information available in the WOStats component or the WOStatisticsStore object on HP-UX.

Workaround: None

Reference: 82143

Problem: Error when WOStatisticsStore object is reset in mid-session

Description: WOApplication allows you to set the WOStatisticsStore object with the method **setStatisticsStore:**. If you invoke this method and a session has already begun, an error occurs. WOSession's **appendToResponse:inContext:** method stores the current WOStatisticsStore object in a temporary variable, then invokes the action method (which in this case results in the WOStatisticsStore being released), and then sends a message to the now invalid WOStatisticsStore.

Workaround: Only use **setStatisticsStore:** in your application's **init** method (or constructor).

Dynamic Elements Known Problems

Reference: 82172

Problem: Exception raised "inside a FORM" causes WOActiveImage to render wrong

Description: If your application catches an exception that happens to have been thrown inside of a dynamic element that is wrapped in <FORM> tags, the exception causes WOActiveImage to render as "INPUT type=image". Your application could remain in a bad state until someone successfully submits a form.

Workaround: Override WOApplication's **handleRequest:** method to do the following:

```
// Add this to spoof the compiler since WOHTMLStaticForm is private
@interface WOHTMLStaticForm {
}
+ (void)setInForm:(BOOL)aFlag;
@end

// Add this to your subclass of WOApplication

- (WOResponse *)handleRequest:(WORequest *)aRequest
{
     // Force the state of "WOHTMLStaticForm" to be
     // correct at the start of each request.
     [WOHTMLStaticForm setInForm:NO];
     return [super handleRequest:aRequest];
}
```

Reference: 82178

Problem: Removing an element from an array bound to a WORepetition causes array out of bounds

Description: If the **list** attribute of a WORepetition (or any other dynamic element with a **list** attribute) is bound to a mutable array, it's possible to alter the size of the list while WORepetition is iterating over it. WORepetition expects the **list** to remain the same size for the duration of the iteration.

Workaround: If you expect to alter a mutable array that is being used as the **list** attribute, you should make an immutable copy of the array and bind the immutable copy to the WORepetition.

Reference: 64906

Problem: WOCheckbox **checked** attribute doesn't work as expected

Description: WOCheckBox's **checked** attribute should return YES or NO as specified in the API, but it actually returns **self** or **nil**.

Workaround: Test the checked attribute for **nil** or non-**nil** values instead of NO or YES. You may also want to write your own checkbox component that returns YES or NO instead of **self** or **nil**.

Reference: 78766

Problem: Container tags in WOConditional and WORepetition don't work as expected

Description: If you have HTML like the following in your template:

```
<TABLE>
<TR>
      <TD>
            <WEBOBJECT NAME=anItem></WEBOBJECT>
      </TD>
      <!-- want to make a three column table -->
      <WEBOBJECT NAME=isThird>
      </TR><TR>
      </WEBOBJECT>
</TR>
</TABLE>
```

WebObjects will not properly place the TR tags to close the rows as expected.

Workaround: Replace the '</TR><TR>' with a WOString:

```
CloseRowTags: WOString {
      value = "</TR><TR>";
      escapeHTML = NO;
}
```

Java Support Known Problems

Reference: None

Problem: Java applications fail if CLASSPATH is defined incorrectly

Description: If you have a CLASSPATH environment variable defined, you may receive an error similar to the following:

```
Errors parsing script file 'C:\TEMP\JavaTest1\Application.java': Scripts of type 'Java'
cannot be automatically mapped to a Class object in the WebObjects application runtime. Such
scripts must be compiled, first.
```

When you launch a WebObjects 3.5 application that uses Java, the CLASSPATH is built up dynamically from your environment's CLASSPATH and internally computed CLASSPATHs. If your environment does not have a CLASSPATH defined, WebObjects will add *NeXT_ROOT*/**NextLibrary/Java** and *NeXT_ROOT*/**NextLibrary/JDK/lib/classes.zip** to the internally computed CLASSPATH. These two paths are required for a WebObjects Java application to run.

If you have a CLASSPATH environment variable dfined, these two paths are not added to the internally computed path; therefore, if you define a CLASSPATH, you must include these two paths in order for your WebObjects applications to run.

Workaround: Prepend the following to your definition of CLASSPATH:

```
NeXT_ROOT/NextLibrary/Java;NeXT_ROOT/NextLibrary/JDK/lib/classes.zip;
```

where *NeXT_ROOT* is the directoy in which you installed WebObjects software.

Reference: 74941

Problem: **Session.wos** and **Session.class** crashes your application

Description: Having both a **.wos** and a **.class** file for the same class (such as **Session.wos** and **Session.class**) will cause your application to crash.

Workaround: Remove the **.wos** file.

Reference: 79559

Problem: Array on WORepetition gets messaged after being deleted in a Java application

Description: If you have a WebObjects application written in Java and one component has this declaration:

```
WORepetition {
      list=myMethod;
      ...
}
```

myMethod is called and creates a next.util.ImmutableVector, which is passed through the Java bridge. If the array is not referenced from Java code, it becomes a candidate for deletion by the Garbage Collector. The Objective-C side does not know this, so the WORepetition tries to access the object after it is deleted.

Workaround: Explicitly reference the array returned by **myMethod** in the component's **.java** file, ensuring that the Garbage Collector is not going to touch it as long as the component lives.

```
private ImmutableVector _myVector;
ImmutableVector myMethod() {
      if (_myVector==null) {
            // compute _myVector here
      }
      return _myVector;
}
```

Reference: 80965

Problem: Bridged objects are never deallocated

Description: Bridged Java/Objective-C objects instantiated in Java are never deallocated by the Java Garbage Collector.

Workaround: For large-scale deployments, implement the **ForceGarbageCollection** method in **Application.java**. See the **Application.java** file in any of the examples in

<*DocRoot*>/WebObjects/Examples/Java.

Reference: 81694

Problem: Throwing an exception in a static class initializer crashes Java applications

Description: Throwing an exception from within a static class initializer like the following:

```
public class Application extends WebApplication {
      static
      {
            ... throw new Exception(...);
      }
```

often causes the application to crash without reporting the thrown exception.

Workaround: Don't throw exceptions in static initialization blocks.   Instead, log an error to the console and exit.

Reference: 80242

Problem: JavaWrapper projects log errors about headers during build.

Description: When building a relatively simple JavaWrapper project, you can get more than 4000 error messages coming from headers included by **Foundation.h** (for example, **winnt.h**). None of these errors result in a build failure.

Workaround: Before building, go to the Project Build panel, click the check mark to open the Build Options panel, and select the "Continue after error" option. Alternatively, you can change the build target to **install** (on the same panel) to avoid seeing these errors altogether.

Reference: 81714

Problem: The Java utility class next.util.PropertyListUtilities was removed from *NEXT_ROOT*/**NextLibrary/Java**

Description: The next.util.PropertyListUtilities class was provided for client-side components to easily handle property lists passed over from the server application. It was removed from the classes found in *NEXT_ROOT*/**NextLibrary/Java**. It still exists under the web server's document root.

Workaround: If you have server-side Java that uses this class, copy the on **.class** file into your project.

Reference: 80393

Problem: Java bridge does not support Distributed Objects

Description: Distributed Object classes do not have equivalents on the Java side of the bridge. This means, for example, that any method that returns NSDistantObject cannot be bridged.

Workaround: None

Reference: 73519

Problem: Constructor helper function doesn't work with floats

Description: It isn't possible from Objective-C to instantiate a Java class using a constructor that contains one or more floats. This includes Java subclasses of Objective-C classes.

Workaround: Type **float** arguments as **double** instead.

Reference: 74127

Problem: Spurious error messages displayed when Java loads libraries on Solaris

Description: **NextObject.loadLibrary()** tries to find the correct library by prepending a number of standard search paths to the supplied library name. For every path that doesn't succeed, **ld.so** generates an error message similar to the following, even if the loading process eventually succeeds.:

```
ld.so.1: /opt/java/bin//../bin/sparc/java: fatal: JavaTest: can't open file:
errno=2 (JavaTest)
```

Assuming that the load process does eventually succeed, these spurious messages can be ignored.

Workaround: None.

Reference: 75000

Problem: You still need to know about Objective-C selectors when messaging from Java

Description: Some of the methods in the **next.util** package require that you know Objective-C selector names. The following code excerpt illustrates one such situation:

```
myVector = new MutableVector();
// fill the vector here
myVector.sortUsingMethod("compare:");
```

Workaround: None.

Reference: 75284

Problem: #import includes files multiple times in **bridgit** on Windows NT

Description: If you have a header file that imports the same file more than once using different names, such as `#import "Foo.h"` and `#import "d:/Headers/Foo.h`", it will be parsed by **bridgit** multiple times, and methods declared in the imported files will be treated as multiple declarations. This results in duplicate methods being emitted in the generated Java classes.

Workaround: None.

WOApplication Known Problems

Reference: 82267

Problem: Overriding **createSession** breaks active sessions count, refusing new sessions

Description: The default implementation of **createSession** now keeps track of the active sessions count and checks to see if the WOApplication is refusing new sessions before it creates one. If you override **createSession**, you must perform these activities in your implementation as well.

Workaround: You only need to override **createSession** if you want to name your session object something other than Session. In that case, add the following lines inside your method:

In Objective-C :

```
- (WOSession *)createSession {
      WOSession *sessionInstance;

      if ([self isRefusingNewSessions]) {
            [self takeValue:YES forKey:@"_refuseThisRequest"];
      }
      [self takeValue:[self activeSessionsCount]+1 forKey:@"_activeSessionsCount"];

      /* Here instantiate your own session object */
      sessionInstance = [[[MySession alloc] init] autorelease];
      if (!sessionInstance) {
            [self takeValue:[self activeSessionsCount]-1 forKey:@"_activeSessionsCount"];
            [NSException raise:NSGenericException
                  format:@"%@: Unable to create new session.", self];
      }

      return sessionInstance;
}
```

In Java :

```
public Session createSession() {
      MySession sessionInstance;

      if (this.isRefusingNewSession()) {
            this.takeValueForKey(true, "_refuseThisRequest");
      }

      this.takeValueForKey(this.activeSessionsCount() + 1, "_activeSessionsCount");

      /* Here instantiate your own session object */
      sessionInstance = new MySession();
      if (sessionInstance == null) {
            this.takeValueForKey(this.activeSessionsCount() - 1, "_activeSessionsCount");
            System.out.println("Unable to instantiate session.");
      }

      return sessionInstance;
}
```

Reference: 81204

Problem: **terminateAfterTimeInterval:** doesn't work if application is not "touched"

Description: The WOApplication method **terminateAfterTimeInterval:** requires at least one request to be handled within the specified time interval for the application to shut down.

Workaround: None

Reference: 81630

Problem: **refuseNewSessions:YES** may cause endless redirects if only one instance is running

Description: If an application has programmatically been set to refuse new sessions and the application is run without using Monitor and only a single instance of the application is running, it causes endless redirects. The application redirects the request back to the WOAdaptor, which tries to pick another instance, but since there is only one instance running, it will pick that instance again.

Workaround: Use the Monitor application to deploy the application. Monitor will detect that there are no non-refusing instances running and return an error page.

Resource Management Known Problems

Reference: 81524

Problem: On Windows NT, WOResourceManager returns path with /.

Description: The path returned by WOResourceManager's **pathForResourceNamed:inFramework:** contains both \ and / characters.

Workaround: None

Reference: 81995

Problem: Java ResourceManager method **pathForResourceNamedInFramework** returns empty string instead of **null**

Description: In some rare circumstances, the Java version of the ResourceManager method **pathForResourceNamedInFramework** returns an empty string when it should be returning **null**.

Workaround: Test for both **null** and **""** to know if a resource has been found.

Client-Side Components Known Problems

Reference: 81792

Problem: Netscape 4.03 appears to hang when using client-side components

Description: There is a bug in Netscape 4.03 related to socket creation on the local host. Because of this bug, client-side components appear to hang when submitting their first request. This bug also affects the Web Assistant in Direct To Web.

Workaround: Use another version of Netscape (down to 3.0) or Internet Explorer.

Reference: 73905

Problem: Client-side components do not work with Netscape Navigator 3.01

Description: A bug in Netscape Navigator 3.01 prevents the use of client-side components in that browser.

Workaround: For client-side components, use any version of Netscape Navigator numbered 3.0 or higher with the exception of 3.01 or 4.03.

Reference: 81767

Problem: WOApplet can specify the archive file for AppletGroupController

Description: You can now specify the archive (**.jar**) to be used for the AppletGroupController class for client-side components. You do so in a manner similar to their regular applets' archives.

For example:

```
MyApplet: WOApplet {
    :
    archive = whatever;
    archiveNames = "some,names";
    agcArchive = "theArchiveForTheAgc";
    agcArchiveNames = "some,names, forTheAgc";
    :
}
```

NOTE: Beware the limitations of your browser; you generally can only use a single archive file per page.

Reference: 82249

Problem: **agcArchive** can only be specified once per page

Description: Netscape Navigator version 4 on all platforms has the following limitations:

· It only supports one **.jar** file in the **archive** attribute of the <APPLET> tag

· When two applets on the same page are trying to communicate using an interface class (as is the case when you use client-side components, which use a class named AppletGroupController), Netscape insists that this interface class be found in the same place (that is, the same **.jar** file or regular **.class** file) on both sides. Otherwise it refuses to recognize that they are the same class.

Workaround: The only way for client-side components to use **.jar** files in Netscape 4.0 is the following:

1. Create one **.jar** file, *X***.jar**, that contains:

    - Your Java client-side classes
    - The **next.wo.client** package
    - Any third party packages you are using

2. Include the binding **archive="***X***.jar"** in all the WOApplets on the page.

3. Include the binding **agcArchive="***X***.jar"** in one of the WOApplets on the page.

This ensures that all the classes needed by both your applets and the AppletGroupController will be found in the **.jar** file. Incidentally, this is probably also how you will get maximum performance out of **.jar** files.

Reference: 78197

Problem: Snapshotting does not work when caching is turned off

Description: A WOApplet will not snapshot the component values when component-definition caching is disabled (that is, the -c option was not specified on the command line, or the message **[WOApp setCachingEnabled:NO]** was sent). This means that a request initiated by a client-side component will result in a response that will contain all the key-value pairs even though some of these pairs may not have changed during the request. This may have some side effects. You should make sure that your client-side component behaves the same way when caching is turned on (since snapshotting will work when caching is turned on).

Workaround: Turn component-definition caching on in order to enable snapshotting

State Management Known Problems

Reference: 66834

Problem: WebObjects applications that store state in the server are not indexed well by internet indexing robots

Description: When indexed by the robots of internet indices like AltaVista, WebCrawler, or Yahoo, WebObject applications that keep state in the server will have many entries (sometimes hundreds, depending on the robot's crawling algorithm). For example, the "Surfshops" page of the CyberWind examples would show up under URLs such as these:

http://XXX/cgi-bin/WebObjects/Examples/WebScript/CyberWind.woa/
12345678901234567890/Main.wo/12345/1.0/1/XXX
http://XXX/cgi-bin/WebObjects/Examples/WebScript/CyberWind.woa/
12345678901234567890/Main.wo/12345/1.2/1/XXX
http://XXX/cgi-bin/WebObjects/Examples/WebScript/CyberWind.woa/
12345678901234567890/Main.wo/12345/1.4/1/XXX
: : : : : :

Clicking the links corresponding to these URLs will typically not generate useful results, since the server state the URLs refer to has usually timed out by the time the link is clicked. If server state is not being timed out in a particular application, the many hits an indexing robot can generate may lead to that application being overwhelmed by the state it has to keep.

This problem only occurs with applications that keep state in the server. Also, for an application to be found by an indexing robot, some URL in that application has to be referenced, directly or through intervening pages, by a page that is submitted for indexing.

Workaround: To prevent indexing of all WebObject applications on a site, create a file called **robots.txt** in the document root of the site's web server, and put lines like the following in the file:

```
# Dissallow robots from indexing all WebObjects apps.
User-agent: *
# All WebObjets URLs on this site start with this.
Disallow: /cgi-bin/WebObjects
```

To prevent indexing of particular applications, use a **robots.txt** file similar to this one:

```
# Dissallow robots from indexing certain WebObjects apps.
User-agent: *
# Dissallow indexing of WebObjects examples.
Disallow: /cgi-bin/WebObjects/Examples
# Dissallow indexing of one app that has server state.
Disallow: /cgi-bin/WebObjects/MyApp
```

In your version of **robots.txt**, replace "/cgi-bin" by the path of your cgi-bin directory (for example, "/Scripts").

If an indexing robot indexes your WebObjects applications in spite of a **robots.txt** file, or for more information, see **http://info.webcrawler.com/mak/projects/robots/robots.html**.

Reference: 76516

Problem: "Backtracked too far" message when using frames

Description: In frames-based applications, the first few pages that the application generates will often stay visible to the user for long periods without being regenerated. It's possible for those pages to be bumped from the page cache (default size = 30). If the user views 30 or more pages in some frame and then tries to interact with a page in another frame, they will get the "Backtracked too far" message. This is because the page was bumped from the page cache while the user viewed lots of other pages.

This is generally not an issue with non-frames applications since there's little likelihood that a user will backtrack a large number of pages (e.g., 30). Even in frames applications, to encounter this problem, the user must view a large number of pages without touching one of the pages in the other frames, and this doesn't happen very often.

Workaround: Set the page cache larger (using the WOApplication method **setPageCacheSize:**) to minimize probability that this will occur.

WebScript Known Problems

Reference: 81973

Problem: WebScript retains intermediate values in expressions as they're evaluated

Description: When evaluating a nested expression like **[[SomeClass alloc] init]**, WebScript retains and autoreleases the intermediate value returned from **alloc**. If this isn't the value eventually returned from **init** (as is the case with all bridged Java objects and a few classes in Foundation), WebScript ends up autoreleasing an object that was never initialized and that might even have already been deallocated.

Workaround: Avoid using **alloc/init** in WebScript. If the class you are instantiating has a +*className* method available (which performs an **alloc**, **init**, and **autorelease**), use it instead. If the class does not have a +*className* method, use the **new** method (as in **[SomeClass new]**). Always use the **new** method to instantiate Java classes from WebScript.

Reference: 67852

Problem: WebScript cannot talk to NSProxies

Description: If you obtain a proxy in WebScript and try to send messages to the proxy, you will get the error message "Unknown type 16."

Workaround: Use Objective-C instead of WebScript.

Reference: 82270

Problem: Modern WebScript syntax causes infinite recursion in some cases

Description: If you do the following, WebScript goes into an infinite recursive loop:

```
- init {
    // BAD CODE!!!! Causes infinite loop
    super.init;
        :
    return self;
}
```

Workaround: Use parentheses to denote that **init** is a method..

```
- init {
    super.init();
      :
    return self;
}
```

Reference: 73867

Problem: Variable argument lists don't work in modern WebScript syntax.

Description: You can't use methods that have a variable number of arguments (such as **logWithFormat:** or NSString's **stringWithFormat:**) in the modern syntax.

Workaround: Use the Objective-C syntax style. That is, instead of:

```
self.logWithFormat(@"I am %@", self);
```

use this:

```
[self logWithFormat:@"I am %@", self];
```

Reference: 79411

Problem: WebScript categories don't work unless component-definition caching is turned on

Description: If you define a category in a component's **.wos** file, you will receive an error at runtime like the following:

```
Errors parsing script file
'/NextLibrary/WebServer/htdocs/WebObjects/MyWebStuff/MyApp.woa/MyComponent.wo/
MyComponent.wos' into class
'WOScriptedClass(/WebObjects/MyWebStuff/MyApp.woa/MyComponent.wo/MyComponent)' with
```

```
superclass 'WOComponent':
Exception while executing statement :
      - methodInCategory {
          ...
```

where **methodInCategory** is a method defined in the category. The problem arises because the script file is read more than once when component-definition caching is turned off. The first time the script is read, the category method is added to the runtime. The second time the script is read, the runtime considers the method a duplicate and produces an error.

Workaround: Turn on component-definition caching with the **-c** command line option or by sending the WOApplication message **setCachingEnabled:YES** during the application's **init** method.

Examples Known Problems

Reference: 81829

Problem: The WebObjects examples are not distributed built on HP-UX.

Description: The WebObjects examples are not distributed in built form on HP-UX as they would take up too much disk space.

Workaround: Build any examples that you want to run by hand.

Reference: 81875

Problem: The directories in the WebObjects examples should be writable by root for Mach, Solaris, and HP-UX

Description: When the directories aren't writable, you can get unpredictable errors.

Workaround: Become root and recursively change the permissions of the **<*DocRoot*>/WebObjects/Examples** so that owner has write permissions.

Reference: 78914

Problem: FDFJava: Exceptions raised when loading some PDF pages.

Description: When WebObjects loads certain PDF pages, the following exception is raised:

```
PDF exception, expecting x fields, found y.
```

This exception occurs because the PDF parser included in WebObjects 3.5 expects only optimized PDF files.

Workaround: Ensure that all PDF files present in **.wo** directories are optimized. You can do this for a given PDF file by opening it in Acrobat Exchange and using the Save As command.

Reference: 75464

Problem: DodgeLiteJava may throw an exception when selecting a car color.

Description: When running the DodgeLiteJava example, if you are selecting a car color and you click outside the active areas of the map (these areas do not perfectly correspond to the color samples) the example will throw an exception.

Workaround: None.

Reference: 81547

Problem: Permissions on SessionStore example do not allow file to be written

Description: For the FileSessionStore example to work, the **SessionArchives** directory must be writable. It is installed with no write permissions on all platforms except Windows NT.

Workaround: Change the permissions on the SessionStore example so the **SesionArchives** directory is writable.

Documentation Known Problems

Reference: 81882

Problem: On Windows NT, OpenStep Books Online file points to obsolete information.

Description: The OpenStep Books Online file points to out-of-date Enterprise Objects Reference material.

Workaround: WebObjects, Enterprise Objects, and Foundation documentation for WebObjects 3.5 is distributed in HTML and accessible through the WebObjects Home Page. You can access the WebObjects Home Page through the WebObjects program group in the Start menu. If you're developing OpenStep applications, some of the information in OpenStep Books Online still applies--namely the Application Kit Reference.

Reference: 81443

Problem: WODynamicElement class specification example is incorrect

Description: The example shown in the WODynamicElement class specification will not work properly if the dynamic element is used on a page that contains client-side components. If the request is generated from a client-side component, the output is not HTML, and thus, the dynamic element's **appendToResponse:inContext:** method should not append HTML to the response.

Workaround: Use the **isFromClientComponent** method to determine where the request originated.

In Objective-C:

```
- (void)appendToResponse:(WOResponse *)response inContext:(WOContext *)context {
      if ( ![[context request] isFromClientComponent] ) {
            // append HTML to the response
      }
}
```

In Java:

```
public void appendToResponse(Response response, Context context) {
      if (!context.request().isFromClientComponent()) {
            // append HTML to the response
      }
}
```

Reference: 81220

Problem: Documentation describes an insecure method of debugging applications.

Description: The documentation describes the preferred method of debugging WebObjects applications, which is to place the project directory under the document root of the web server. If you use a web server that is accessible by outside users, placing the project in the web server's document root represents a security risk; you've exposed your source code to the outside world.

Workaround: If security is an issue,   don't place your source code in the document root. One workaround is to place your project directory under **NextLibrary/WOApps** and debug applications from there.

Reference: None

Problem: Some documentation is not installed on your system.

Description: We are currently working on the following pieces of documentation. When they are ready, they will be available through the Apple website (http://devworld.apple.com/technifo/techdocs/enterprise/WebObjects):

·€ Enterprise Objects Framework reference in Java
·€Direct to Web documentation
·€More complete documentation for WebObjects Builder and Project Builder (titled *WebObjects Tools and Techniques*)

Workaround: Periodically check the "Updated docs at Apple" link on the WebObjects Home Page.

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