

WebObjects Release Notes

For Release 3.1

Last Updated 5/30/97

This file contains notes about the WebObjects family of products. WebObjects release 3.1 is the current release for the Window NT, Solaris, and NEXTSTEP platforms.

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 + CGI Adaptor (Windows NT Workstation or Server and UNIX platforms)
- Netscape + NSAPI adaptor (Windows NT Workstation or Server and UNIX platforms)
- IIS Server + CGI Adaptor (Windows NT Workstation or Server) (install as 65135 and 78035)
- IIS Server + ISAPI adaptor (Windows NT Workstation or Server) (install as 65135 and 78035)
- Website 1.1D and greater (Windows NT Workstation or Server)
- Purveyor (Windows NT Workstation or Server)

- NCSA, CERN, Apache, + CGI (UNIX platforms)

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 NeXT about a bug's status.

Installation with Microsoft IIS Server

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 NeXT 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 NeXT 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 NeXT software, click the Browse button and use the panel that opens to select a root directory (for example **C:**).
- 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: 78035

Problem: On Windows NT, machd and nmserver are not installed properly if you pick an installation directory like **C:**.

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 if the installation directory is at the top level (**C:**). 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
```

Installation General

Reference: 78749

Problem: (Gamma release only) Installation on NT 3.51 fails: OpenStep applications do not start.

Description: Two of the environment variables that WebObjects sets contain invalid information. This invalid information consists of partial strings (derived from the BOOT.INI file or the registry hierarchy containing the equivalent information) and appended to the end of the PATH and NEXT_ROOT environment variables.

The Pasteboard and Window servers (and possibly **machd** and **nmserver**, as well) rely on the ability to run certain executables (such as **make_services**) which they can't find because of the corrupt environment variables. Among other things, OpenStep apps (such as WebObjects Builder, EOModeler, and TextEdit) won't launch because they're unable to contact the WindowServer (this error will be logged into the Application event

log, which is viewable via EventViewer).

Workaround: In the System control panel, edit NEXT_ROOT and PATH so as to remove any trailing garbage, then reboot. This bug is fixed in the FCS release of WebObjects 3.1.

Reference: 78798

Problem: Installation corrupts the PATH system variable on certain installations of NT 4.0.

Description: The Windows NT installer changes the registry entry of the PATH system environment variable from REG_EXPAND_SZ to REG_SZ. This may prevent any resolution of environment variables contained in the PATH value (like \$SYSTEMROOT), thereby hiding many executables.

Workaround: Reset the path to REG_EXPAND_SZ. On the Environment tab of the System control panel, select the PATH system variable. Place your cursor over the Value field and click the left mouse button. Click Set, then Apply. Reboot your system for the change to take effect. It's a good idea to perform this step even if you don't see any negative results.

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: 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 white space.

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

Reference: 73668

Problem: PDO Apache installation conflicts with PDO WOF installation.

Description: The Apache installation script wants to install Apache to **/usr/NextLibrary**. WOF 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 WOF. Note that you must know what the cgi-bin and document root will be ahead of time, as the WOF installation script prompts you for this information.

Reference: 74723

Problem: Enterprise Objects examples wrongly included.

Description: The WebObjects package erroneously includes some examples based on the Application Kit and Enterprise Objects UI layer (Customer, AssociationExample, AssociationPalette, Inventory, PointOfSale, Studios). These examples will not compile unless you have **AppKit.framework** and **EOInterface.framework**, which are part of the OpenStep product.

Workaround: Ignore these examples.

Reference: 78544

Problem: WOF Enterprise Deployment on Mach: Java directory not copied to document root.

Description: On Mach, the installer does not copy **/NextLibrary/Java** into the **/NextLibrary/WebServer/htdocs/WebObjects/Java** directory. This directory is required under the document root so that Java applets (client-side components) can access the Next Java classes they use.

Workaround: Do the following as root:

```
cp -rf /NextLibrary/Java /NextLibrary/WebServer/htdocs/WebObjects/Java
```

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: 78743

Problem: Uninstalling WebObjects for Mach removes Project Builder makefiles.

Description: WebObjects installation overwrites the Project Builder makefiles provided with the OPENSTEP release with updated makefiles that WebObjects requires. If you uninstall WebObjects, these makefiles are removed, but the previous versions are not put back in place, so you have no Project Builder makefiles.

Workaround: Before uninstalling, copy the directory **/NextDeveloper/Makefiles/pb_makefiles** to a safe location. After the uninstallation is complete, copy the directory back to **/NextDeveloper/Makefiles/pb_makefiles**.

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 NeXT 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 NeXT 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 NeXT software, click the Browse button and use the panel that opens to select a root directory (for example **C:**).
- 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.

WebObjects Adaptors

Reference: 64517

Problem: Netscape HTTP servers on NT do not accept cgi programs with an .exe extension

Description: For the CGI server adaptor to work with the Netscape servers, the CGI adaptor (that is, the WebObjects program in the server cgi-bin directory) can't have any extension. Other NT HTTP servers may require the '.exe' extension.

Workaround: Add the .exe extension to the WebObjects program if your Windows NT HTTP server requires it.

Reference: 73392

Problem: Microsoft's IIS server: Autostarted applications can't be killed.

Description: You can't use TaskManager to kill autostarted applications on Windows NT with the IIS server. By default, the IIS server runs autostarted applications as a special user with no privileges. Consequently, you don't have permission to kill these processes.

Workaround: Do one of the following:

- Go to your WWW server configuration panel and change the anonymous user to your user name and password. Be careful to type the password correctly as you are not warned if you enter the wrong password. Reboot your computer for the change to take effect.
 - If you want to keep the anonymous user for security reasons, start your WebObjects application by hand. During application development starting by hand is generally the better course, so you can observe debugging messages in the terminal window.
-

Reference: 67984

Problem: Apache Server on HP-UX: Can't change user to "nobody"

Description: By default, the Apache server sets the user to nobody when launching cgi-bin processes. However, the UID of nobody is -2, which causes setuid to complain about an invalid argument.

Workaround: Change the nobody UID and nogroup group ID in **/etc/passwd** and **/etc/group** to positive numbers.

Reference: 69715

Problem: Cannot disable autostarting functionality.

Description: There is no way to turn off the autostarting functionality of a WebObjects application stored under the document root. That means that outside users may be able to run undesired applications, such as installed examples.

Workaround: Before deploying an application, clean the site of any undesired applications. To disallow autostarting of an application, store it in *NeXT_Root/NextLibrary/WOApps*, and rename the application executable. If the application is scripted, rename **WODefaultApp** to any other name. If the application is compiled, rename the executable so that its name is different from the **.woa** name. (See also 73208)

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: 72341

Problem: Adaptor mode no longer supported

Description: WebObjects 2.0 had a server HTTP adaptor mode that allowed you to contact a specific instance of an application using URLs of the following format:

`http://.../WebObjects/AppName:instanceNumber@hostName`

In WebObjects 3.0 and higher, the instance number is an optional attribute that the user can no longer specify.

Workaround: None.

Reference: 73208

Problem: Clients can browse source code within *Document_Root* **WebObjects** unless you take steps to prevent them.

Description: Users could view the source code of your WebObjects applications unless you deny them access or move your application. For example, if someone enters this URL

`http://your_server/WebObjects`

Workaround: Either of these two approaches will protect your source from prying eyes:

1. Configure your web server to disallow read access to WebObjects-related files (*.wos, *.wod, *.plist, etc.). This won't affect the operation of WebObjects applications, which access the file system directly, but will prevent the server from dispensing these

files to browsers.

2. For WebObjects 3.0 and higher, move your WebObjects applications from the server's document root to *NeXT_ROOT/NextLibrary/WOApps*.

If you use the second approach, you must move the entire application (the ".woa" directory) to *NeXT_ROOT/NextLibrary/WOApps*. However, if you have any statically linked resources such as sound or image files (for example, , you must leave a sparse copy of your application under the document root. In this case "sparse" means that the application's directory structure is reproduced in the document root, but the only files it contains are the static resources that the server must dispense to a client's browser.

If you choose to move a scripted application to *NeXT_ROOT/NextLibrary/WOApps*, you must take one further step. Each application in the **WOApps** directory must contain an executable, but a scripted application located in the server's document root relies on the default application *NeXT_ROOT/NextLibrary/Executables/WODefaultApp* (or **WODefaultApp.exe** on Windows NT) as its executable. So, when you move the scripted application into **WOApps**, copy the default application into your scripted application, and then rename this copy with the name of your application.

Be sure to keep these two copies in sync at all times.

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 **/tmp** on UNIX platforms and **c:\temp** on Windows NT). 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:

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

7 Restart your server.

7 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: 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.next.com 3000
```

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

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

Workaround: None

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: 69158

Problem: A running WebObjects application may erroneously complain that a new page or component is missing

Description: If you run a WebObjects application, and then add a new component to the application, and then try to access the new component's page in the browser, you receive a message saying the application cannot find the HTML template for the new component, even though the **.wo** directory and the **.html** template file both exist.

Workaround: Restart the WebObjects application.

While you are building and debugging a WebObjects application, you shouldn't use the autostart capabilities of the WebObjects adaptor. Instead, run the application from a shell and then kill and restart the process as pages and components are added.

WebObjects Builder

Reference: 69669

Problem: WebObjects Builder does not support languages other than WebScript.

Description: WebObjects Builder automatically generates a **.wos** file even if the component's code is being written in another language.

Workaround: Manually delete the **.wos** file within the **.wo** directory, then add the code file you need.

Reference: 73162

Problem: Consistency Check assumes unknown names in WebScript are Class references.

Description: The Consistency Check command assumes any unknown name to be a reference to a Class name. For example, in

```
[MyClassName someMethod];
```

since MyClassName is not known, it is assumed to be a Class reference. Class references are resolved at run time, so the Consistency Check command does not inform you of the unknown reference.

Workaround: None.

Reference: 72229

Problem: Browse button in Preference panel doesn't select directories on NT.

Description: If you click on the Browse button in the Preferences panel to specify the document root, the Open panel doesn't select directories.

Workaround: In the Open panel, select a file inside the directory you want to select then click OK. After the Open panel is dismissed, edit the text field to remove the file name and leave only the directory name.

Reference: 69976

Problem: There is no way to use a WOString for a title

Description: Because the document's title is set in the inspector, there is no way to use a WOString for the title.

Workaround: Open the **.html** file directly, create a WOString within the document's title, then set its bindings appropriately in the **.wod** file.

Reference: 70765

Problem: Can't add an empty list.

Description: If no text is selected, the Format->List->New List menu item doesn't do anything.

Workaround: Type a few characters, select them, then choose Format->List->New List; the characters appear as part of a list. You can now select the characters and delete or type over them. Alternatively, drag a list from the Static Elements palette to produce an empty list.

Reference: 71070

Problem: Selecting inserted HTML element and typing replaces element.

Description: Drag a heading off the Static Elements palette into a component, and then go to the component window and start typing. The whole element is deleted, rather than just the

contents.

Workaround: Before typing, make sure only the text within the heading is selected; if the selection goes beyond the heading (including either of the new-lines before and after the heading), the heading will be deleted.

WebObjects Builder: HTML Support

Reference: 68067

Problem: Pasting RTF does not work very well

Description: Sometime, spaces and attributes are lost when you paste RTF text into WebObjects Builder.

Workaround: None.

Reference: 68068

Problem: Pasting RTFD does not work

Description: Attachment (images) are not preserved.

Workaround: None.

Reference: 68816

Problem: Implementation of anchors is incomplete.

Description: WebObjects Builder supports local anchors but does not provide any way to edit them besides custom markers and generic inspectors.

Workaround: Use the custom HTML marker to create the anchor and edit it in that marker's inspector.

Reference: 69495

Problem: It is difficult to see a form's boundaries.

Description: It is not always obvious if you are editing inside or outside of a form.

Workaround: To see the form's boundaries, click near one of the controls, and a dashed line will appear. At this point, you can be sure that the editing will take place inside the form.

Reference: 71374

Problem: Adding a form after setting text attributes occasionally creates bad display.

Description: Enter some text, select it, and increase the font size. Now drag in the form from the palette. The fields will initially be in the big font from before. Select the text and make it smaller. The display of the whole form is wrong now.

Workaround: When you first encounter the bad display of the form, choose Edit->Undo from the menus until the form displays properly. Now select the entire form and choose Format->Text->Plain Text from the menus. You should now be able to set the font attributes within the form as you wish.

Reference: 72504

Problem: Horizontal line appears inside a heading.

Description: Drag a heading from the Static Elements palette, edit it, then drag a horizontal line from the palette. The line ends up inside the title.

Workaround: Delete the horizontal line, and then move the text selection to the beginning of the line following the heading. (The dotted line around the heading should disappear.) Now drag the horizontal line from the palette; it will appear outside of the heading.

Reference: 72550

Problem: Cut and paste from part of a list does not work

Description: If you copy part of a large list, then paste outside of a list, the structure of the list is lost. Instead, you get only the text from the list without bullets or carriage returns.

Workaround: Select the entire list, copy and paste it, and then remove the pieces you do not want from the new copy.

Reference: 73282

Problem: Cannot create a reusable component with no form in it.

Description: In WebObjects Builder, it's impossible to create a component that has form elements in it but does not have a surrounding <FORM> tag.

Workaround: Edit the **.html** manually to remove the <FORM> tags.

Reference: 73541

Problem: Existing components that have two dynamic elements with the same name confuse WebObjects Builder.

Description: Some legacy WebObjects applications have components that contain references to two dynamic elements with the same name; the **.wod** file then had a single entry for that name, which both dynamic elements shared. While this worked under WebObjects 2.0, it was never supported; all dynamic elements within a document should have unique names. If such a document is opened and saved inside of WebObjects Builder, WebObjects Builder produces an unusable document.

Workaround: None. Edit the component's HTML file manually to remove the duplicate reference.

Reference: 73653

Problem: Cannot change the top-level marker from <BODY> to another marker.

Description: The HTML expert inspector does not allow you to change the top-level marker (the <BODY> marker). This is particularly important when trying to create a page with frames, where the top-level should be <FRAMESET>.

Workaround: None. A <FRAMESET> page must be created manually.

Reference: 73726

Problem: Dragging a color onto selected text does not work well if the selected text has mixed fonts.

Description: If the selected text has mixed fonts (different font sizes or different font attributes like bold and italics), dragging a color from the Color Panel to the selection does not always work; sometimes only part of the selected text gets the color, instead of all of it.

Workaround: Drag to the color well in the toolbar instead of to the selection.

WebObjects Builder: Image Support

Reference: 69412

Problem: Cannot create image maps

Description: There is no way to create image maps in WebObjects Builder.

Workaround: For client-side image maps, create the map description using the generic HTML tag on the Static Elements palette. Then select the image, open the inspector, and click HTML Expert in the inspector. In HTML Expert mode, you can add the attribute necessary to reference your map. For server-side image maps, edit the attributes of the image as above, then use a text editor to create the map file.

Reference: 73574

Problem: Resources in untitled documents don't work very well.

Description: If you have an Untitled document (that is, a document that has never been saved), manipulating resources from WebObjects Builder does not work properly. For example, if you create an active image and then set the button title in the inspector, the image in the document is not updated.

Workaround: Save the document before adding resources so that there is a location in the file system where resources can be manipulated.

WebObjects Builder: Inspectors

Reference: 70071

Problem: Sometimes the structure selection in the text is out of sync with the inspector.

Description: When you type, the smallest structure around the insertion point is selected (gets the dotted rectangle), which is not necessarily the same as the structure selected in the inspector.

Workaround: In the inspector, click the icon of the structure you want to select (even if it is already highlighted); the selection drawn in the main window will update to match the inspector.

Reference: 70117

Problem: Undo doesn't work while editing classes.

Description: If you are changing a type in the Classes window and select Undo without pressing return to end editing in the inspector, the Classes window ends up in an invalid state, and potentially the underlying types are set incorrectly as well.

Workaround: Press return before selecting Undo.

Reference: 73370

Problem: Bindings occasionally go to the wrong element when double-clicking.

Description: If you have a dynamic element imbedded in another dynamic element and you try to bind a variable to the outer element by selecting the outer element and double-clicking the variable, sometimes the binding erroneously goes to the inner element. This can occur when the outer dynamic element contains nothing but the inner element. That is, there

are no characters between the start of the outer element and the start of the inner element or between the end of the inner element and the end of the outer element.

Workaround: Open the inspector and select the bindings inspector for the inner web object. Delete the erroneous binding. Now select the bindings inspector for the outer web object and create the desired binding. Alternatively, put a character between the beginning of the outer web object and the beginning of the inner one; double-clicking will work as expected. Later, delete the extra character.

Reference: 73485

Problem: HTML Expert inspector is easily confused

Description: If the inspector is left in HTML Expert mode, it occasionally loses track of the current document selection. Once this has occurred, the display of the expert inspector can become wildly inaccurate, and the expert inspector is no longer useable.

Workaround: Switch the inspector to its normal mode then back to the HTML Expert mode; this should synchronize it with the current document selection. However, the HTML Expert mode should only be used briefly, to make a small change directly to the HTML.

Reference: 73569

Problem: WOBrowser inspector is misleading.

Description: WOBrowsers have inspectors to allow you to set the options within the browser, but this is only relevant if you are creating a purely static SELECT. A WOBrowser will ignore them.

Workaround: None.

Reference: 73805

Problem: Binding the **string** attribute of a WOHyperlink does not remove the body of the WOHyperlink in the HTML

Description: Add a WOHyperlink to a component, and then bind its **string** attribute to anything. The word "Hyperlink" is not removed from within the WOHyperlink, and when the application is run, the link shows up with the word "Hyperlink" followed by the value of the **string** attribute.

Workaround: Delete the word "Hyperlink" between the WOHyperlink's icons in the document.

Reference: 73891

Problem: Binding inspector occasionally does not display all bindings.

Description: If the binding inspector is open, and new bindings are created without using the inspector (for instance, by double-clicking a variable), the binding inspector occasionally does not display the newly-created binding.

Workaround: Click another button in the inspector's path, then go back to the binding inspector; this should update the binding inspector's display. If this does not work, try closing and reopening the inspector.

WebObjects Builder: Palettes

Reference: 68866

Problem: Placing form objects outside of a form generates a new form.

Description: Currently, if you place controls individually on a page without placing a form object first, you get a separate <FORM>...</FORM> around each control.

Workaround: None.

Reference: 69491

Problem: Form Elements palette is not complete.

Description: There are many form functions that cannot be accessed from the palette, including adding a submit or reset button, creating a completely empty form, or combining several form elements into a single form.

Workaround: Use the entries in the Format->Form menu, which provides all of the above functionality.

WebObjects Builder: Application Windows

Reference: 72429

Problem: Save As is not enabled for the application window

Description: You cannot use File->Save As to rename an application.

Workaround: Save the application and all components within it. Then copy it in the file system.

Reference: 74247

Problem: WebObjects Builder has no direct interface to Project Builder.

Description: Unlike Interface Builder, WebObjects Builder currently does not notify Project Builder when application documents are edited. In addition, WebObjects Builder does not currently provide users with a way to add resources to Project Builder's PB.project file.

Workaround: Use Project Builder to add resources to web-based projects.

WebObjects Builder: Script Window

Reference: 73647

Problem: Find panel only works in HTML display of component window

Description: The Find panel only searches the main component window's HTML display. You cannot use the Find panel in the script window.

Workaround: To perform searches in the script window, use the keyboard equivalents (Command-e to enter the selection, Command-g to Find Next, Command-d to Find Previous). Note that on Windows NT, the key equivalents are Control-e, Control-g, and Control-d, respectively. There is no workaround for doing Find/Replace operations in the script window.

Reference: 75485

Problem: WebObjects Builder generates "protected Object foo[]" for an array

Description: When you create a new array variable in Java, WebObjects Builder creates an array of type Object instead of an ImmutableVector. You should get an ImmutableVector, because

an array is usually created to work with WORepetitions.

Workaround: None.

WebObjects Builder: Dynamic Elements Support

Reference: 70141

Problem: WOStrings should pick up font attributes.

Description: If you make a WOString bold, the visible contents of the WOString do not display in bold. This makes it difficult to see the font attributes on a WOString.

Workaround: Click within the WOString and look at the toolbar. The highlighted buttons correctly portray the font attributes of the string.

Reference: 72313

Problem: Attributes get lowercased when copied or when put on a palette.

Description: When you copy and paste or drag an element from the palette, all of the attributes that appear in the binding inspector will be forced to lowercase letters. In certain circumstances, this may cause duplicate keys to appear in the binding inspector.

Workaround: Choose attribute names that are all lowercase.

Reference: 74059

Problem: WebObjects Builder adds quotes around constant values.

Description: If you use the bindings inspector to set an attribute's value to a constant, save the application, and then close and reopen it, WebObjects Builder adds quotation marks around your constants. If you save the component again, the quotation marks are written out to the component's **.wod** file.

In many cases, this is harmless. For example, if you bind the **multiplesubmit** attribute of WOForm to the constant 0, it will eventually be written to the **.wod** file as

```
multiplesubmit = "0"
```

When the framework evaluates **multiplesubmit**, it will send **intValue** to the string "0", which returns the integer value 0.

However, this is a problem if you use defined constants such as YES and NO. If you set **multiplesubmit** to NO in the Builder and it is written to the **.wod** file as

```
multiplesubmit = "NO"
```

The framework sends **intValue** to the string "NO", which does not return 0.

Workaround: Always use 1 and 0 in places of YES and NO.

WebObjects Builder: Database Integration

Reference: 73566

Problem: Dragging and EOModel from a **.woa** onto itself deletes the EOModel.

Description: If you have previously added a WODisplayGroup by dropping an EOModel file on your

.woa, you will have a copy of that EOModel file in your **.woa**. If you subsequently drop that EOModel file onto the same **.woa**, WebObjects Builder tries to recopy the EOModel but deletes the file in the process.

Workaround: You can move the file outside of the **.woa** first before copying or copy from the original place that the EOModel file came from.

Reference: 74399

Problem: If eomodel is a link, WOBuilder copies the link, not the original file.

Description: This is only a problem on UNIX systems. If the user selects a symbolic link to an eomodel file, rather than the original file itself, the symbolic link is copied into the component. This is particularly troublesome when working with a relative link, since the link will not be valid in the new location.

Workaround: Find and select the original file, rather than the link.

Reference: 78072

Problem: Using the "Set Adaptor Info" menu item in EOModeler to change the connection string results in a model that can't connect to the Oracle database.

Description: If you have a connection dictionary that includes both the "serverID" and "hostMachine" keys and you use the "Set Adaptor Info" menu item to set the "serverID" value to a something new, the old hostMachine key remains in your connection dictionary. Unfortunately, this remnant will lead the OracleEOAdaptor to construct a connection string of the form "T:<hostMachine>:<serverID>" rather than just using "<serverID>". This is likely to happen when you configure the examples to use a local database.

Workaround: You can remove the "hostMachine" key from the connection dictionary using the EOModeler ConnectionDictionary Inspector. Select the root EOModel, and display the Inspector. You can then select the row with the "hostMachine" key and click "Remove" to clear it from the connection dictionary.

WebObjects Builder: Language Support

Reference: 75922

Problem: If your language preference (as set in the Preferences application) is set to something other than Japanese, it's not possible to enter Japanese characters.

Description: On OPENSTEP for Mach, most applications automatically switch fonts if you type a character that is unrepresentable in the current font (for example, typing a Japanese character into TextEdit when Helvetica is the current font). WebObjects Builder doesn't do this, however.

Workaround: When you run WebObjects Builder with a language preference set to something other than Japanese and want to type Japanese characters, you have to manually set the font in Builder's Preferences panel to a Japanese font before entering the Japanese text.

WebObjects Framework

Reference: 66089

Problem: WebObjects applications on Solaris cannot access environment variables

Description: If you autostart a WebObjects application on Solaris, the process is owned by user nobody. nobody has no user environment, and thus does not have access to the **NEXT_ROOT** environment variable, meaning it won't be able to locate NeXT resources files (for example, time zone files).

Workaround: Do one of the following:

- Install NeXT software under the root directory.
- Change your web server's configuration so that CGI process are launched by a user with the appropriate environment set up rather than user nobody. This will allow you to autostart web applications from the browser.
- If you want to keep the nobody user for security reasons, start your WebObjects applications by hand.

Reference: 73419

Problem: Project Builder does not add **.woa** extension

Description: When you create a project of type WebObjectsApplication in Project Builder, Project Builder does not append the extension **.woa** to the name you specify. WebObjects cannot recognize a WebObjects application without the **.woa** extension.

Workaround: Load **WebObjectsSupport.bundle** as described in "Creating a Compiled Application" in the book *Getting Started With WebObjects*.

Reference: 78366

Problem: **gdb** breaks on unknown signal

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

Workaround: Use the **continue** command to continue execution when **gdb** breaks. To prevent this from happening in the first place, do one of the following :

7 Before running your application in **gdb**, enter the command **set signal-exception 4**.

7 Create a **gdb.ini** file containing the **set signal-exception** command and save it in each of your **.woa** projects. You also can put it in your home directory to affect all projects that you work on.

Reference: 75080

Problem: Applications that use a **WebObjects.conf** don't respond when run under **gdb**.

Description: An application that works correctly when autostarted from a Web browser may not respond to requests when started from within **gdb**. This problem only occurs when you are using a **WebObjects.conf** file to specify a port number for the application to listen on.

This problem occurs on both Windows NT and on Solaris.

Workaround: None

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 **`setBacktrackingEnabled:`** method.

Reference: 73745

Problem: Externally editing `.woo`, `.wod`, or `.wos` files created by WebObjects Builder can cause conflicts

Description: If you rename an instance variable using an editor other than WebObjects Builder on a component created by WebObjects Builder, it can cause your application to become non-functional. At run time, you receive a message that the variable does not exist. The problem is that the information in the `.woo` file (a file maintained internally by WebObjects Builder and the WebObjects Framework) is now out of sync with the file you edited.

Workaround: Always use WebObjects Builder to edit the script file. Do not edit the `.wod` file. If you encounter the error message, delete the `.woo` file.

Reference: 75264

Problem: You can't specify the `.woa` extension when invoking an app from the command line

Description: When starting an application from the command line, WebObjects appends a `.woa` extension to the application name you specify, whether or not you've supplied an extension. So for instance, if you try to invoke your app as follows:

```
$ ./Zowee -d c:/netscape/server/docs Zowee.woa
```

It won't start because WebObjects won't be able to find an application directory named

"Zowee.woa.woa".

Workaround: Don't specify the **.woa** extension when starting WebObjects applications from the command line.

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: 77932

Problem: On PDO platforms, scripted applications that access a database won't run.

Description: For the PDO platforms, you need to explicitly link in the database client libraries. This isn't done for you automatically, so you can't create a totally scripted application that accesses a database.

Workaround: Create a "compiled" application and make sure that the Makefile includes the appropriate client libraries. The application can still contain scripted components, but you need to create and compile your own **main()** function. See "Creating a Compiled Application" in the book *Getting Started With WebObjects* for instructions on how to create a compiled application. The DodgeDemo example illustrates this process for an application accessing data from an Oracle database.

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.

WebObjects Framework: Request Handling

Reference: 71990

Problem: Cannot cancel a **WODefaultApp** using the ORACLE adaptor

Description: If you run have a WebObjects application that uses the ORACLE adaptor and you run it from the command line, you cannot use Control-C or Control-Z to kill the executable. The Oracle client library traps all signals, including Control-C and Control-Z, so the executable never receives the signal.

Workaround: Use **kill -9** to kill the executable.

Dynamic Elements

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: 66845

Problem: Imagemap files can only reside in the **.wo** directory

Description: Image map files used with WOActiveImage are assumed to reside in the **.wo** in which they are referenced. No indication is given by WOActiveImage when it cannot find this image map file.

Workaround: Make sure the imagemap file is saved in the **.wo** in which it is referenced.

Reference: 73112

Problem: WebObjects Builder includes <HTML>, <HEADER>, and <BODY> tags in reusable components

Description: WebObjects Builder inserts <HTML>, <HEADER> and <BODY> tags in all HTML templates. If you use WebObjects Builder to create a reusable component, WebObjects will generate them the <HTML>, <HEADER>, and <BODY> tags into the parent page at the reusable component's location, meaning that a page might have any number of <HTML>, <HEADER>, and <BODY> tags. Netscape Navigator and Microsoft Internet Explorer browsers ignore the redundant tags. However because this is illegal HTML, there is no guarantee that all browsers will behave properly.

Workaround: Outside of WebObjects Builder, remove the redundant tags from the reusable component's HTML template file.

Client-Side Components

Reference: 73905

Problem: Client-Side Components may not work with some browsers

Description: Java Client-Side Components in WOF 3.0 have been tested with both Netscape Navigator 3.0 and Internet Explorer 3.0. Earlier versions or prereleases of these

browsers are not guaranteed to work with Client-Side Components. In particular Beta releases 5 and 6 of Netscape Navigator 3.0 had Java run-time bugs that prevent the proper functioning of Client-Side Components.

Workaround: None.

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.

Reference: None

Problem: WOApplet now has **target** attribute.

Description: In WebObjects 3.1, the WOApplet dynamic element has a **target** attribute so that client-side components can return a page in a specified frame. Use the **target** attribute to specify the name of the frame in which the page should appear.

WebObjects Framework: HTML Generation

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 `<` or `>`, respectively, causing the script to fail on the client.

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

Reference: 78170

Problem: WebObjects parser replaces the single quote (') in the JavaScript statement with a double quote (")

Description: If you have included a Java Script statement surrounded by single quotes as the value of an attribute for an HTML tag, the WebObjects parser erroneously replaces the single quotes with double quotes.

Workaround: Put the Java Script code in a function and use the function reference as the value of the attribute.

WebObjects Framework: State Management

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 apps since there's little likelihood that a user will backtrack a large number of pages (e.g., 30). Even in frames apps, 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.

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/CyberWind@xxx/
Main.wo:1.395502469.0$Main.2.3.8.0.1
http://XXX/cgi-bin/WebObjects/Examples/CyberWind@xxx/
Main.wo:1.395502469.2$Main.2.3.8.0.1
http://XXX/cgi-bin/WebObjects/Examples/CyberWind@xxx/
Main.wo:1.395502469.4$Main.2.3.8.0.1
: : : : :
```

Clicking on 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:

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

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

```
# Disallow robots from indexing certain WebObject apps.
User-agent: *
# Disallow indexing of WebObject examples.
Disallow: /cgi-bin/WebObjects/Examples
```

```
# Disallow 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 apps in spite of a **robots.txt** file, or for more information, see <http://info.webcrawler.com/mak/projects/robots/robots.html>.

Reference: 72951

Problem: CookieSessionStore size limitations

Description: Due to the size limitation on cookies and the limit on environment variables for the CGI interface on certain systems, the cookie session store is only useable for cookie session state that has less than 2K in size. Note that the developer is still able to use cookies programatically -- this limitation is only on CookieSessionStore.

Workaround: None.

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: 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: 74070

Problem: WebScript sends **retain** too early if you use **alloc/init**.

Description: If you have the following script:

```
- awake {
    id var = [[MyClass alloc] init];
    // something
    [var release];
}
```

the WebScript engine sends the retain message to the object **var** prior to sending **init**. This causes no problems in most cases, but it might cause problems for some classes (for example, Foundation collection classes such as NSArray and NSDictionary).

Workaround: Avoid using **alloc** in WebScript wherever possible.

Java Support

Reference: 73519

Problem: Constructor helper function doesn't work with floats on Solairs

Description: On Solaris, 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: None.

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: 74941

Problem: Session.wos and Session.class crashes the Java bridge

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: 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: 75004

Problem: Project Builder's debugger panel doesn't display Java command line.

Description: To facilitate debugging, Java Web applications display the command line used to launch the Java VM. This message is not displayed in Project Builder when launched in the debugger.

Workaround: Run the application in the shell or modify the **main.m** to log the command line using **NSLog()**.

Reference: 75022

Problem: Attempting to launch an uncompiled Java application can disable your server

Description: If you attempt to launch a WebObjects Java application before it has been compiled, it will cause errors on your web server and will leave the server in a state from which you'll be unable to launch any other WebObjects applications. Essentially, the WebObjects process that's trying to contact the uncompiled application holds on to the WebObjects.autostart file and waits for five minutes, to prevent another concurrent autostart.

Workaround: If this happens, you'll need to kill the WebObjects process as you won't be able to remove the WebObjects.autostart file. Note that you may have to restart the web server in order to unlock the WebObjects.autostart file.

Reference: 75207

Problem: Cannot overload methods in hybrid Java subclass

Description: You cannot overload methods that are in "hybrid" Java classes (a "hybrid" Java class is a Java subclass of an Objective-C class). Thus, for instance, the following code will cause your app to crash when the myMethod is invoked from Objective-C:

```
public class MyComponent extends Component {
    public void myMethod() {...}
    public void myMethod(int anint) {...}
}
```

Workaround: None

Reference: 75467

Problem: Subclasses of **next.wo.Component** shouldn't be given a package name

Description: Subclasses of **next.wo.Component** shouldn't be given a package name, or the **pageWithName()** method won't be able to find them. This is because **pageWithName()** creates the page by looking up and instantiating the component class with the same name as the page. If the Java class is given a package name, the class lookup will fail.

Workaround: None.

Reference: 75519

Problem: **NSClassFromString()** is very slow

Description: Code that makes heavy use of **NSClassFromString()** will run slower once the Java extensions are installed.

Workaround: None.

Reference: 75537

Problem: Java Browser comes up empty

Description: If you installed the latest release of the Java Extensions on top of the prerelease, the JavaBrowser may come up empty. This is because some old versions of the loadable bundles used by the browser will still be present.

Workaround: Traverse to the browser resource directory

(*NEXT_ROOT/NextDemos/JavaBrowser.app/Resources*) and remove the **Decompiler.beandle** and **InputSources.beandle** directories.

Reference: 75466

Problem: EOF client libraries can't be automatically loaded into Java on Solaris systems

Description: On Solaris, the client libraries and adaptor frameworks cannot be automatically loaded by EOF; you must explicitly load a library that is linked against the adaptor frameworks and the client libraries in your Java program. See the **main()** method and Makefile.postamble in the DodgeDemoJava example for the recommended way to do this. Note that some demos--such as MovieDemoJava--won't work unless this change is made.

Workaround: None.

Reference: 75691

Problem: Setting JAVA_HOME produces better error messages during compilation

Description: The JDK environment that's set up by the Java Extensions for compilation needs to have JAVA_HOME set so that the compiler can find the properties file that contains the English error messages. Otherwise, **javac** complains and prints an error message that's only mildly descriptive

Workaround: Using the Windows NT System control panel, create a JAVA_HOME environment variable and set it to *NEXT_ROOT/NextLibrary/JDK-1_0_2*.

Reference: None

Problem: Control-C doesn't always kill **java.exe**

Description: If you use Control-C to kill a running Java application, the **java.exe** process isn't always killed properly. This will cause you to have multiple Java VMs running at the same time.

Workaround: Use the Windows NT TaskManager to ensure that the extra VM processes have been killed off.

Reference: 75284

Problem: #import includes files multiple times in Bridget 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 Bridget 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.

WebObjects Examples

Reference: 74015

Problem: On Solaris, the Performance example requires Solaris Developer

Description: The Performance example requires files (for example, **stdio.h**, **stdlib.h**, **unistd.h**) that are provided on Solaris only if you have Solaris Developer installed. If you do not have

Solaris Developer, compiling the Performance example will fail.

Workaround: Install Solaris Developer before installing WebObjects 3.0.

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: 78442

Problem: DodgeDemo does not work with WebObjects Pro

Description: The DodgeDemo makes use of business objects (Car object). Per the license agreement, the use of business objects is disabled in Pro. Therefore, the example will not work with Pro.

Workaround: None.

Reference: 78446

Problem: **Monitor.woa** doesn't generate pages

Description: The **Monitor.woa** example is missing a **SessionArchives** directory. The example depends on this directory being present. If it's not there, the application sets up the frame set, but doesn't generate any pages.

Workaround: Create a **SessionArchives** directory under the **Monitor.woa** directory.

PDF Extensions

Reference: 78910

Problem: Key paths aren't supported in the PDF extensions

Description: Key paths (for example, **account.number**) are unsupported in this version of the PDF extensions. If you try to bind a PDF field to a variable using a key path, you get either a blank field or this exception:

.. expected N fields and got N-1 ..

Acrobat Exchange treats '.' in field names in a different manner and produces PDF that the WebObject parser does not recognize as a single field.

Workaround: If you need to bind to a variable like **account.number**, declare two accessor methods for it as follows:

```
public String accountNumber() { return account.number; }
```

and

```
public void setAccountNumber(String newValue)
{
    account.number=newValue;
}
```

Bind the PDF field to to the method **accountNumber**.

Reference: 78914

Problem: 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 3.1 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.
