

WebObjects Development Tools Release Notes

For Release 3.5

Last Updated 11/26/97

This file contains notes about the tools that you use to create WebObjects applications:

- Project Builder
- Direct to Web (a feature of Project Builder)
- WebObjects Builder

Also refer to these other release notes:

- WebObjects release notes (known problems with the WebObjects Framework, with installing the WebObjects product, and with deploying WebObjects applications)
- Enterprise Objects release notes (known problems with the Enterprise Objects Framework and the EOModeler application).

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.

Known Problems With Project Builder

In WebObjects 3.5, Project Builder has been modified to better integrate with other tools that you use to write WebObjects applications. The version of Project Builder distributed with WebObjects 3.5 has these known problems. Problems documented OpenStep 4.2 Project Builder Release Note also apply.

Developing OpenStep Applications

When you install WebObjects 3.5, it replaces your existing OpenStep installation with new development tools, frameworks, and files. For example, modifications were made to ProjectBuilder, the makefiles, and project types to accomodate new features only usable in WebObjects. Interface Builder and the AppKit

framework are installed, but have not changed from the OPENSTEP 4.2 release. While we don't believe you should have any problems with developing OpenStep applications, you will not receive any support to help you around problems you encounter. If you need to develop OpenStep applications on a supported platform, use another machine that does not have WebObjects 3.5 installed.

3.1 to 3.5 Project Conversion Facility Known Problems

Reference: 82275

Problem: WebObjects support bundle no longer needed

Description: Previous releases of WebObjects required you to load a bundle named **WebObjectsSupport.bundle** into Project Builder. 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: 79078

Problem: Project Builder conversion of encoded **.wod** or **.wos** files is lossy

Description: When converting an existing WebObjects project that has non-ASCII encodings in either the **.wod** or **.wos** files, Project Builder will convert these files to plain ASCII using a lossy conversion. The files converted in this lossy way are backed up by writing them out with the suffix **.preWOF35** before doing the conversion.

Workaround: None

Reference: 80628

Problem: Project Builder occasionally can try to convert a project more than once.

Description: Under certain circumstances, the upgrade to a project for WebObjects 3.5 can fail to be committed to disk. This may happen if the folder or file is somehow locked by File Explorer (on Windows NT).

Workaround: Ensure the file is writable by Project Builder and not locked in any other program.

Reference: 81293

Problem: Project Builder cannot convert old applications that don't have projects

Description: Prior to WebObjects 3.5, developers were not required to use Project Builder to manage their projects. This is required now, and Project Builder will convert existing directories containing a **PB.project** file to the new format. However, if your project wasn't being managed by Project Builder (and thus has no **PB.project** file), the conversion utility does not apply.

Workaround: Create a new WebObjectsApplication project with Project Builder and manually add all components, classes, resources, images, etc. to the appropriate suitcases.

Reference: 81421

Problem: Project Builder ignores **WOProject.plist** when converting some applications

Description: When converting existing applications to 3.5 projects, Project Builder removes **WOProject.plist** file in some projects (with user intervention) and ignores the file in other projects. This is because the file must have been part of the old project for the conversion utility to know it exists.

Workaround: After converting to WebObjects 3.5, delete the file **WOProject.plist** from all projects as it is no longer required.

Building Projects Known Problems

Reference: 82227

Problem: **install** build target installs into *NEXTROOT*

Description: If you build with the **install** target in Project Builder, the Makefile prepends the *NEXTROOT* to whatever directory you specify as the "Install In" path in the Build Attributes inspector.

Workaround: Before building, click the Build Options button in the Build Panel and the following to the arguments: `DSTROOT=""`

Reference: 81333

Problem: The **debug** target in Project Builder appends unsupported **.debug** extension.

Description: The **debug** target is not supported for WebObjects 3.5 because it appends the

unsupported **.debug** extension (rather than **.woa**) to the application wrapper and the framework binary.

Workaround: Use the target **woapp** or **woframework** and turn off optimizations in the **Makefile.postamble** with:

```
OPTIMIZE_BUILD_CFLAGS =
```

In addition, you may set the DEBUG flag yourself in the following way (in **Makefile.preamble**):

```
OTHER_CFLAGS += -DDEBUG
```

or in the Arguments field of the Build Options panel, add:

```
OTHER_CFLAGS=-DDEBUG
```

Debugging symbols will be available under this scheme.

Reference: 80522

Problem: **(\$NEXT_ROOT)/LocalLibrary/Frameworks** not searched during link process on Windows NT

Description:	On Windows NT, if you use a framework from (\$NEXT_ROOT)/LocalLibrary/Frameworks in another project, the link will fail. This problem may arise if you developed the project on OpenStep for Mach (where the LocalLibrary directory is searched automatically) and then copied it over to Windows NT to recompile.
Workaround:	Manually add (\$NEXT_ROOT)/LocalLibrary/Frameworks to the framework search path in the Build Attributes Inspector.
Reference:	80434
Problem:	Deleting a header that is marked public causes Makefile inconsistency
Description:	If you have a header file in your project that is marked public, it causes a Makefile inconsistency because the header file is removed but the entry in PUBLIC_HEADERS for the header file is not removed.
Workaround:	Make the header file private before removing it.

Java Integration Known Problems

Reference: 81727

Problem: WebObjectsApplication projects with no server-side Java don't build client-side components correctly

Description: When a WebObjectsApplication project has no Java at the top level, the ClientSide Java and Common Java subprojects don't get archived into **.jar** files or put into the application wrapper's WebServerResources directory at all.

Workaround: Add a dummy **.java** file to the Classes suitcase at the top level of your project and rebuild.

Reference: 81717

Problem: Project Builder indexer gets confused over static initializers and static inner classes

Description: Static initializers show up as "-)" in the browser and clicking on them selects the static block as well as the prototype of the next method. The first method in a static inner class shows up as "-;" and clicking it in the browser does the same thing as clicking a static initializer. Some methods invoked within the bodies of methods in static inner classes also show up in the index.

Workaround: None

Reference: 80313

Problem: JavaWrapper projects cannot be subprojects.

Workaround: Make the JavaWrapper project a standalone project.

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.

Miscellaneous Known Problems in Project Builder

Reference: 81640

Problem: Subprojects cannot contain components with the same name

Description: If you have organized your WebObjects application into subprojects, be sure that the component names are unique across the entire project. If two components in two different subprojects have the same name, one component will overwrite the other in the **Resources** directory during a build.

Workaround: None

Reference: 82233

Problem: Project Builder cannot handle project directories with spaces in their names

Description: Project Builder will open projects that are in a directory with spaces in its filename, but whenever another application is used to open a file in the project, Project Builder passes a non-quoted string to the application, which interprets the string as several independent filenames and emits warnings about invalid or non-existent files.

Workaround: Avoid using spaces in project names and project directory names.

Reference: 80912

Problem: The **.woo** part of a component doesn't appear in Project Builder.

Workaround: You shouldn't manually edit a **.woo** file-- it is maintained by WebObjects Builder. Should you need to examine the contents of a **.woo**, you can open it using the File->Open command in Project Builder, and it will appear in Non Project Files.

Reference: 81530

Problem: Unnecessary logs from Project Builder Find Panel

Description: This happens every time you use the find panel:

Sep 25 15:54:05 ProjectBuilder[10525] Could not connect the action setView: to target of class Finder

Sep 25 15:54:05 ProjectBuilder[10525] Could not connect the action setView: to target of class Finder

:
:
:

Workaround: None

Reference: 81337

Problem: Project Builder indexing not updating the user interface

Description: Updates to the user interface seem to stop after a few files get indexed, making it look as if project server is hung.

Workaround: None

Reference: 81687

Problem: Open panels mislabeled

Description: In WebObjectsFramework projects, if you double-click "Web Server Resources" to add a resource, the resulting open panel is named "Add Web Server," not "Add Web Server Resource." If you double-click "Non Project Files," the resulting open panel is named "Add Classes."

Workaround: None

Direct to Web Known Problems

Direct to Web is a new feature in WebObjects 3.5. You create a Direct to Web application by creating a new WebObjectsApplication project in Project Builder and choosing "Direct to Web" as the type of assistance. Project Builder creates a Direct to Web application for you and launches it in your web browser. From there, you can launch the Web Assistant and customize the application. Direct to Web and the Web Assistant have these known problems:

Miscellaneous Known Problems in Direct to Web

Reference: 82130

Problem: Query pages using key-value paths always retrieves all the objects

Description: If you freeze a query page containing a QueryToOneField or a QueryToManyField, the query page or 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: Edit your query component as follows:

In the **.wod** file, there should be one or more WOTextFields, corresponding to the QueryFields:

```
MyField: WOTextField {  
    value=displayGroup.queryMatch.department.name  
}
```

Replace this 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");  
}
```

```
}
```

Reference: 82076

Problem: Direct to Web is slow

Workaround: There are several things you can do to speed up Direct To Web applications:

1. Displaying entities that have a lot of relationships or relationships with a lot of elements on the other side of the relationship is very CPU intensive; use the Web Assistant to hide some of the entities and relationships.
2. Starting the Web Assistant significantly slows down the Direct To Web application.
3. In deployment, make sure you are running with caching on (using the **-c** command line option or the WOApplication **setCachingEnabled:** method). Note that it is currently not possible to generate pages from the Web Assistant with this option turned on.
4. Consider freezing some of the slow pages (especially list pages) from the Web Assistant.

Reference: 82118

Problem: Direct to Web doesn't appear to delete items; they still appear on the list page

Description: It's possible to click the Delete button on the edit page, go to the list page, and still see the object you just deleted. This is because WODisplayGroup does not purge the deleted objects from its list.

Workaround: There are three options:

1) Register to observe the EOObjectsChangedInEditingContextNotification sent by EOEditingContext. When you find an object in the deleted list in the **userInfo** for the notification, remove it from any WODisplayGroups.

2) In the **awake** method for your list page, scan the **displayedObjects** array of the WODisplayGroup and remove any objects not registered in the EOEditingContext.

3) Force the WODisplayGroup in your list page to refetch.

Reference: 81940

Problem: Saving changes in Direct to Web causes error(s)

Description: There are several situations that you can get into where modifying attributes of enterprise objects and then saving will raise exceptions in EOF. These can be due to inconsistencies in your EOModel and your database. They can also be due to improper

modifications to foreign keys.

Workaround: It's important to validate your EOModel (use the consistency checker in EOModeler) and make sure that relationships between entities are mapped correctly.

Reference: 82049

Problem: Direct to Web returns to wrong page after saving changes

Description: If you list some objects of Entity A and inspect a relationship that one of them has to Entity B, edit one of the objects of Entity B, and save changes, you will end up on the query page for Entity B.

Workaround: Direct to Web always returns to the query page for entity that you are editing after you save changes. To change this behavior, generate the component and customize its behavior.

Reference: 80930

Problem: Sorting by first column in Direct to Web produces an error when first column is a relationship

Description: In list pages, Direct to Web by default sorts the list on the first column. If the first column is a relationship, it results in an error.

Workaround: Change the order of the columns with the Web Assistant.

Reference: 82041

Problem: "No query available" message in Direct to Web

Description: Direct to Web displays a "No query available" message when it does not have a way to query the data type for a particular attribute. It does not mean that you cannot query on that entity; it just means that you cannot qualify the fetch based on that particular attribute.

Workaround: You can stop this message from appearing by using the Web Assistant to hide that attribute.

Reference: 81248

Problem: The DisplayImage component can only use images contained in the application itself

Description: The images have to be in the resources of the application for the DisplayImage

component to be able to find them.

Workaround: Add images to the project using Project Builder.

Reference: 82075

Problem: Direct to Web cannot display attributes of type NSData

Description: Direct to Web does not know how to represent attributes of type NSData when generating HTML. NSData objects could be images or other types of data. The raw hexadecimal representation is usually not interesting, so it is not displayed.

Workaround: None

Reference: 81870

Problem: NSDecimalNumbers display differently in **userPresentableDescription** output than in other places in Direct to Web

Description: If an enterprise object uses its **userPresentableDescription** method to display its value in Direct to Web and that enterprise object contains an NSDecimalNumber value, the NSDecimalNumber is formatted using the Money formatter. If that same number is

displayed elsewhere in Direct to Web, it's just a basic float style number with two decimal places of precision. This happens because **userPresentableDescription** picks a default formatter for each attribute based on settings in the EOClassDescription.

Workaround: Don't use **userPresentableDescription**. Display a different key for your enterprise object and set the appropriate formatter for that key.

Reference: 82180

Problem: Date formatters other than the default cause problems in Direct to Web

Description: Certain date formatters can cause Direct to Web to have problems. First, you may not be able to save any changes to the attribute or it may save an empty value. Second, Direct to Web may raise an exception when trying to generate an edit or query page because the formatter can't be parsed.

Workaround: There is no workaround for the first problem. You can workaround the second problem by using the Web Assistant to create a valid date formatter before you get to a edit or query page.

Web Assistant Known Problems

Reference: 81971

Problem: Cannot start the Web Assistant on Solaris.

Description: If you try to start the Web Assistant for a Direct to Web application on Solaris, you receive a log similar to the following:

```
Oct 16 02:31:45 d2walpha[4181] waiting for requests...
ld.so.1: ./d2walpha: fatal: libnet.so: can't open file: errno=2 (libnet.so)
java.lang.UnsatisfiedLinkError: no net in shared library path
    at java.lang.Runtime.loadLibrary(Runtime.java)
    at java.lang.System.loadLibrary(System.java)
    at
    at java.net.ServerSocket.<init>(ServerSocket.java:57)
...
```

Workaround: Make sure **libnet.so** is in your path

Reference: 81189

Problem: When reloading a page that shows the Web Assistant, Web Assistant appears twice

Description: If you display a page in Direct to Web, display the Web Assistant for that page, and then

click the browser reload button, two Web Assistants are displayed.

Workaround: The Web Assistant lives in the bottom frame; you should only reload the top frame (right-click on that frame).

Reference: 81758

Problem: Backtracking with the web browser's Back button confuses the Web Assistant

Description: When you backtrack in the web browser, the Web Assistant does not track the displayed page correctly.

Workaround: Use the navigation buttons in the Direct to Web page instead of using the web browser's buttons.

Reference: 81866

Problem: If the Cancel button is clicked, the Web Assistant loses track of the browser page

Description: If you load a page in Direct to Web and you cancel the load by clicking either the Cancel button on the page or the web browser's Back button, the Web Assistant loses track of the page.

Workaround: Don't use the Cancel or the Back button when the Web Assistant is active.

Reference: 81929

Problem: **userPresentableDescription** is listed as an attribute to query on

Description: When you inspect a to-one relationship in the Web Assistant, the Web Assistant lists the object's attributes that you can use to query the relationship.
userPresentableDescription is erroneously listed as one of these attributes. If you select it, the Web Assistant will perform the query using the primary key.

Workaround: Select another attribute in the Web Assistant.

Reference: 81763

Problem: Web Assistant may overwrite existing component files

Description: When you choose to generate a component using the Web Assistant, the Assistant does not check to see if the component already exists, and it does not prompt you to make sure that you really want to overwrite any existing component with the same name.

Workaround: Make sure that you aren't overwriting anything you want to keep before you choose to

generate a page.

Reference:	81800
Problem:	Generated component not displayed automatically
Description:	When you generate a component for a previously dynamic page in Direct to Web and then click the Reload button in the web browser, the page that is displayed is still the dynamically generated page.
Workaround:	Re-navigate to the modified page using the Direct to Web navigation tools to display the correct page.

WebObjects Builder Known Problems

The version of WebObjects Builder shipped in version 3.5 has these known problems:

The .woo File

WebObjects Builder might add a *Component.woo* file to your component's directory. It does so under these circumstances:

- If you add a WODisplayGroup to the component and you change the configuration of that WODisplayGroup (for example, if you change the sort ordering or any other option in the Display Group Options panel).
- If you want to localize the component's HTML and you've specified a character encoding other than the default.

The **.woo** file is a property list and is formatted like any other OpenStep property list. Because WebObjects Builder and WebObjects expect the **.woo** file to appear in a certain format, you should never edit the **.woo** file directly. If you do so and you introduce a syntax error, WebObjects Builder will not be able to open your component.

HTML Editing in Graphical Display Mode Known Problems

Reference: 82033

Problem: WebObjects Builder attempts to copy resources that don't need copying.

Description: If you have a resource in your project directory that is not currently part of your project, dragging that file into WebObjects Builder has confusing results. WebObjects Builder attempts to copy the file into the project, but first it moves aside the old copy by appending a "~" to its name. The copy fails because the file that WebObjects Builder is trying to copy no longer exists, and you're left with the original file renamed to have a "~" at the end.

Workaround: Add the file to the project using Project Builder and then drag it into WebObjects Builder.

Reference: 82184

Problem: Adding form elements inside a WOForm results in additional FORM tags

Description: Adding a submit button inside a WOForm results in a new FORM around the button in addition to the button itself.

Workaround: Remove the FORM tags in source editing (raw) mode and save while still in source editing mode. Use a FORM rather than WOForm permanently or temporarily.

Reference: 82185

- Problem: Using FRAMESET in your component is tricky
- Description: By default, WebObjects Builder adds <HTML> and <BODY> tags to the beginning and end of every document, so creating a FRAMESET is tricky.
- Workaround:
1. Mark the component as a partial document in the Inspector panel.
 2. Enter the HTML as follows (using generic HTML markers or "raw" source editing mode):

```
<FRAMESET ROWS=50%, 50%>  
    <FRAME SRC="http://www.apple.com" NAME=One>  
    <FRAME SRC="http://devworld.apple.com" NAME=Two>  
    <NOFRAMES>  
        <BODY>You are out of luck..</BODY>  
    </NOFRAMES>  
</FRAMESET>
```

Reference: 82309

- Problem: Dragging files into tables doesn't work
- Description: An image or component dragged into a table will not appear.
- Workaround: Add the desired object using the appropriate toolbar button and set the **filename** and **framework** bindings accordingly. As an alternative, drag the item to a location outside of

the table and then cut and paste it into the table.

Reference:	80154
Problem:	Percentage-sized items in an unsized table cell mess up display
Description:	An unsized table cell claims to have an infinite width to accommodate its contents, and the percentage-sized item (such as an <HR> tag or a table with any percentage width set) demands a fraction of that infinite width, which is still infinite, and so the table cell attempts to expand to accommodate that demand, which ends up scrunching the rest of your table into hardly any space at all.
Workaround:	Set explicit widths for your table columns. In general, WebObjects Builder's table editing code works best with explicit pixel widths set for table cell widths.
Reference:	81756
Problem:	Inspector doesn't update when in structure editing mode on a nested table
Description:	Open a file with nested tables (one table appears in one of the cells of the other table) in WebObjects Builder. Click in the cell of the outer table that contains the inner table. Now click in the inner table, then go to structure editing mode on the inner table (you can

use the button on the toolbar). Bring up the inspector and start selecting different cells within the table. The inspector path doesn't update, making it impossible to inspect the attributes on the table cells and rows of the inner table.

Workaround: When you need to inspect the table cells and rows of the inner table, use content editing mode instead.

Reference: 80839

Problem: Hard to merge two identical, neighboring structures

Description: If you have two identical structures (such as paragraphs, headers, or lists) that are next to each other, it is very difficult to combine them into a single structure. Placing the cursor at the beginning of the second structure and pressing backspace doesn't work; neither does placing the cursor at the end of the first structure and pressing delete.

Workaround: Select the contents of the second structure, choose copy, then go to the end of the first structure and choose paste; this should append the contents of the second structure to the first. Now you can delete the second structure.

Reference: 81344

Problem:	When <PRE> nodes are destroyed or converted, their contents are not properly converted
Description:	If you have a preformatted paragraph (that is, one that uses the <PRE> tag) and use WebObjects Builder's inspector to change it to a normal paragraph (<P>), any carriage returns, tabs, or multiple spaces in the contents of the paragraph are not stripped or replaced by markers; they are put directly into the contents of the new paragraph. This destroys the WYSIWYG rendering of the HTML; newlines will appear within the new paragraph, even though they would not appear in a browser.
Workaround:	Switch to source editing mode then back to graphical mode; this will remove the extraneous whitespace. You can now add in newlines and other formatting wherever desired.
Reference:	81346
Problem:	Centering several list items at once splits the list
Description:	If you center a single list item, the list item properly places a paragraph around its text and centers the paragraph, leaving the list itself intact. However, if you select several list items and center them, the list will be split at the selection boundaries, and the selected pieces will be centered.

Workaround: Center each list item independently.

Reference: 82098

Problem: Difficult to put a list around a dynamic element

Description: If you select some text that includes a dynamic element, then click the List button on the Structures toolbar, the dynamic element will not be wrapped by a list item, but will appear directly within the list Ⓓ probably not what was desired.

Workaround: Create the list first, then cut and paste the dynamic element into the list at the desired location.

Reference: 81533

Problem: Dragging in a shared component places it around the current selection

Description: If you have text selected in WebObjects Builder and you drag in a shared component, WebObjects Builder attempts to surround the text with the shared component, rather than replacing the text.

Workaround: Make sure no text is selected when you drag in a shared component. If desired, delete

any text that you wanted to replace after you drag in the component.

Reference:	82016
Problem:	Backspace does not delete shared components
Description:	You cannot delete a shared component using the Backspace key.
Workaround:	Select the entire component by clicking one of the icons at the beginning or end of the component, then press the Delete key.

HTML Editing in Source ("Raw") Mode Known Problems

Reference:	81162
Problem:	Whitespace in between WEBOBJECT tags may be ignored
Description:	WebObjects Builder does not treat whitespace immediately inside the WEBOBJECT start and end tags correctly. It ignores such whitespace when reading in files, and it uses its own rules to determine how much whitespace to display. Normally, such whitespace is not significant (according to the HTML specification).

Workaround: Use a non-breaking space (- entered using shift-space in WebObjects Builder) instead of a space when you want to force whitespace. Manually edit your HTML prior to actual deployment to fix whitespace issues.

Reference: 82063

Problem: WebObjects Builder pretty-printing wraps text, which results in erroneous spaces

Description: If you create the following component outside of WebObjects Builder:

```
<BODY>
Hello <WEBOBJECT NAME=MyString></WEBOBJECT>, how are you?<BR>
</BODY>
```

Then open the component in WebObjects Builder and saved it, WebObjects Builder generates:

```
<BODY>
Hello
<WEBOBJECT NAME=MyString></WEBOBJECT>
, how are you?<BR>
</BODY>
```

which creates an unwanted space before the comma.

Workaround: Turn off pretty-printing in the preferences panel.

Reference: 80803

Problem: Mismatches between a component's **.html** and **.wod** files do not cause parse errors switching out of source editing (raw) mode

Description: Entries in the **.wod** file that do not have a matching WEBOBJECT tag are silently removed when you switch from source editing (or raw) mode to graphical display mode.

Workaround: None

Working With WODisplayGroups Known Problems

Reference: 81556

Problem: Display groups cannot be reset

Description: Before you configure a display group variable, a dash appears next to the variable's name in the object browser. After you configure the display group, a checkmark appears

next to it, and there is no way to reset the variable back to a dash.

Workaround: Delete the display group variable in the source code file and then save the file. Switch back to WebObjects Builder and add a new display group variable.

Reference: 82105

Problem: Double-clicking doesn't open the Display Group Options panel.

Description: In WebObjects Builder, you double-click a display group variable to open the Display Group Options panel. On Windows NT under rare circumstances, WebObjects Builder can get in a state where double-clicking a display group variable isn't recognized.

Workaround: Triple-click the variable.

Bindings Known Problems

Reference: 80253

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

Description: WebObjects Builder lets you write illegal values (such as values with unrecognized

characters) for binding attributes and values. These values are written to your **.wod** file without validation, so your **.wod** file will not be correct.

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

Reference: 81517

Problem: The Select Image button doesn't use the **filename** and **framework** attributes.

Description: If you add an image to a component using the Select Image button, the WOImage **src** attribute is used. If the image is part of your project or it is part of a framework linked to your project, you should use the **filename** and **framework** attributes instead of **src**.

Workaround: Change the attributes to be correct, or drag the image instead of using the Select Image button.

Reference: 81688

Problem: Cannot add two images with the same name in WebObjects Builder

Description:	If your project contains an image named Foo.gif and you drag in an image named Foo.gif from a framework onto a component, WebObjects Builder will not allow you to add that image to your component.
Workaround:	Add conflicting images in Project Builder and set the proper framework and filename bindings in WebObjects Builder's WOImage inspector.
Reference:	81674
Problem:	WebObjects Builder does not offer filename and framework attributes for WOBody or WOEmbeddedObject
Description:	The dynamic elements WOBody and WOEmbeddedObject support the filename and framework attributes; however, their Inspector panels in WebObjects Builder do not list these attributes.
Workaround:	Add them by hand.

Project Builder Interaction Known Problems

Reference:	74247
------------	-------

Problem:	WebObjects Builder and Project Builder do not notify each other when application files are edited
Description:	WebObjects Builder does not notice any changes made in Project Builder to source code files, HTML files, or .wod files until you save. Likewise, Project Builder does not notice files that you edit in WebObjects Builder until you save.
Workaround:	Save the edited files before switching applications.
Reference:	None
Problem:	WebObjects Builder does not display variables or methods beginning with an underscore
Description:	If you declare a variable or method with a name that begins with an underscore (_), WebObjects Builder's object browser does not display that variable or method. WebObjects Builder considers any variable or method name beginning with an underscore to be private API that you don't want displayed in the object browser.
Workaround:	None.
Reference:	80517

Problem: WebObjects Builder parser does not recognize modern syntax methods

Description: Methods declared using this syntax:

```
function myMethod() {  
    some.modern().looking().code();  
}
```

will not be properly represented in the object browser.

Workaround: Use classic syntax when declaring methods in WebScript:

```
- myMethod {  
    some.modern().looking().code();  
}
```

Reference: 82123

Problem: Saving a new component into a project does not parse the component's code file.

Description: Application and session keys are not displayed until you edit the component's source code file.

Workaround: Edit the component's source file, either by adding variables/methods from WebObjects

Builder or by switching to Project Builder.

Reference: 81964

Problem: No keys/actions displayed for classes containing inner classes

Description: No keys or actions are displayed for Java classes that contain inner classes. This is a Java 1.1 feature that the WebObjects Builder parser doesn't support.

Workaround: Don't use inner classes in component source files. If you must use inner classes, you can type in any missing keys using the WebObjects Builder inspector.

Reference: 82216

Problem: Using Objective-C keywords as method names can crash WebObjects Builder

Description: Naming an Objective-C method **break**, **default**, or any other reserved word can crash WebObjects Builder when it parses the source file.

Workaround: Don't use reserved words as method names.

HTML Graphical Display Known Problems

Reference: 81889

Problem: Some unknown images appear blank

Description: When WebObjects Builder encounters an image whose source URL cannot be loaded and whose size is configured to either shorter or narrower than the unknown image icon typically used, it will display the image as an empty box the size requested by the image description.

Workaround: None

Reference: 82099

Problem: Images do not give any indication that they're loading

Description: When you insert a large image into WebObjects Builder, there is a noticeable delay before the image is displayed. No indication is given that the image is being loaded. Since the image is loaded in a background thread, the application continues to be useful during this time, giving the impression that the insertion of the image failed somehow.

Workaround: None

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 source editing (raw) mode. For server-side image maps, edit the attributes of the image in the Inspector, then use a text editor to create the map file.
Reference:	82154
Problem:	Host lookup can hang WebObjects Builder for up to 5 minutes
Description:	On Windows NT systems that have DNS lookup enabled but are not currently on a network, WebObjects Builder can hang while attempting to load components that contain URLs with non-local (<code>http://...</code>) references.
Workaround:	Disable DNS lookup temporarily or open a Bourne shell window and enter the following command:

```
defaults write WebObjectsBuilder DisableHostLookup YES
```

and then restart WebObjects Builder.

Reference: 80654

Problem: WebObjects Builder doesn't understand % escape sequences in URLs

Description: When WebObjects Builder reads in HTML that includes URLs that use % escape sequences, it does not properly interpret the escape sequences. This prevents WebObjects Builder from properly accessing files referenced by the URLs; in particular, if the URL specifies the location of an image, the image will not be loaded. Likewise, when asked to construct a URL for a particular file location, it will not properly escape characters that are illegal in URLs. This most commonly occurs with filenames that include spaces; the spaces should be translated to %20, but are not.

Workaround: There is no workaround for the misreading of URLs; WebObjects Builder will simply not be able to access those URLs. However, any time WebObjects Builder creates a URL that is not properly escaped, you can edit that URL (either in an inspector or in source editing mode) to contain the proper escape sequences; WebObjects Builder will properly remember the new URL.

Reference: 80641

Problem: Font size displays incorrectly when insertion point is next to an image or other attachment

Description: When the cursor is right next to an image or any other kind of attachment (anything that isn't ordinary text), the font size control displays incorrectly. Usually, it jumps to the largest size. When you type, however, the typing appears at the appropriate size.

Workaround: None

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.

Using Palettes Known Problems

Reference: 74847

Problem: Changing the **codebase** attribute breaks client-side components

Description: WebObjects Builder shows the codebase attributes of client-side components as editable. However, if you change the value of this attribute, the client-side component won't work because it requires the classes in **<DocumentRoot>/WebObjects/Java**.

Workaround: Don't change the **codebase** attribute.

Reference: 80825

Problem: The Make Editable item in the palette menu is actually a Toggle Edit Mode

Description: The Make Editable menu item does not do exactly what it's description implies -- it functions more as a Toggle Edit Mode button.

Workaround: None

Reference: 81805

Problem: There is no Save As command for palettes.

Description: There is no way to save a palette to a new location.

Workaround: Copy the palette in Workspace Manager or File Explorer.

Reference: 81835

Problem: Palette items cannot be renamed on Mach

Description: Palette items created cannot be renamed in WebObjects Builder on Mach.

Workaround: Open the palette on an NT machine to do the renaming or manually edit the palette's plist file.

Miscellaneous WebObjects Builder Problems

Reference: 81367

Problem: WebObjects Builder cannot find the WebObjects Home Page

Description: If the document root isn't set up correctly, WebObjects Builder does not display WebObjects Home Page when you choose the Help command.

Workaround: Set the document root in the Preferences panel. Choose Tools->Options to display the Preferences Panel, click the suitcase icon, and type the document root in the Document Root text field.

Reference: 81953

Problem: Documented edited indicator not updated during table modifications

Description: When editing a component containing a table, modifications inside the table are not reflected in the document's edited state (in the window titlebar) until leaving the table.

Workaround: None

Reference: 81461

Problem: Copyrights are out of date and they refer to NeXT rather than Apple

Description: Project template copyrights are out of date and they refer to NeXT rather than Apple.

Workaround: None

1997 Apple Computer, Inc. All rights reserved.

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.