

WebObjects Release 4.0 Post-Installation Instructions

About This Document

This short guide describes steps you should follow immediately after you've installed WebObjects. By following the instructions in this guide, you'll be assured that the WebObjects package was installed properly.

Before reading this guide, be sure you have followed all of the instructions in the printed *WebObjects 4.0 Installation Guide* and that you've checked the *WebObjects Release Notes* and *Enterprise Objects Framework Release Notes*, both of which are available online. Updated documentation is maintained on Apple's web site; go to <http://www.apple.com/webobjects/> and click on the Resources link.

- Post-Installation Steps
 - Windows NT Post-Installation Steps
 - Solaris Post-Installation Steps
 - HP-UX Post-Installation Steps
 - Using Microsoft Access
 - Setting Up the Sample Databases
- Verifying the Installation
- Troubleshooting
 - Checking the Installation
 - Problems With Scripted Applications
 - Problems With Compiled Applications

Windows NT Post-Installation Steps

After you've finished installing on Windows NT, you may want to perform one or more of the following steps:

- Install Yellow Box Services
- Install Monitor as a Service
- Microsoft IIS or PWS Post-Installation Steps
- Obtain and Install Database Client Libraries

Install Yellow Box Services

The Install Shield wizard installs **machd** and **nmserver** and properly configures them as services unless you've changed your *NEXT_ROOT* environment variable to the root directory (which you must do if you're using a Microsoft server). These two services are required to run WebObjects Builder, Project Builder, and EOModeler.

To install these processes as services, open a Bourne shell window and enter the following commands:

```
> cd $NEXT_ROOT/Library/System
> ./machd.exe -install
> ./nmserver.exe -install
```

Install Monitor as a Service

Monitor and MonitorProxy can be installed as services on Windows NT systems. To install Monitor as a service, open a Bourne shell window and enter the following commands:

```
> cd $NEXT_ROOT/Library/WebObjects/Applications/
Monitor.woa
> ./MonitorDaemon.exe -InstallMonitor
```

To install MonitorProxy, use:

```
> cd $NEXT_ROOT/Library/WebObjects/Applications/
Monitor.woa
> ./MonitorDaemon.exe -InstallMonitorProxy
```

To reverse these operations, “uninstalling” them as services, use:

```
> cd $NEXT_ROOT/Library/WebObjects/Applications/
Monitor.woa
> ./MonitorDaemon.exe -UninstallMonitor
```

Or:

```
> cd $NEXT_ROOT/Library/WebObjects/Applications/  
Monitor.woa  
> ./MonitorDaemon.exe -UninstallMonitorProxy
```

Microsoft IIS or PWS Post-Installation Steps

If you're using Microsoft IIS or PWS as your HTTP server, perform the following additional post-installation steps:

Grant Administrator Privileges to the CGI User

- The CGI user must have Administrator privileges to access the NEXT_ROOT environment variable. This is necessary if you want to enable *load balancing* (distributing the processing load among multiple instances of a WebObjects application). Load balancing is a performance feature that you'd use on a deployment site.

To grant Administrator privileges to the CGI user, do the following:

1. Open the "User Manager for Domains" application under Administrative Tools.
2. Double-click the entry with the full name "Internet Guest Account." (The user name usually starts with "IUSR".)
3. In the window that opens, click the Groups button.

You'll see two tables, one called "Member of" and one called "Not member of."

4. Move the Administrators group from the "Not member of" table to the "Member of" table.
5. Click the OK button.

Obtain and Install Database Client Libraries

To use Enterprise Objects Framework on Windows NT, you must have the appropriate database client libraries. The Sybase client libraries are provided on the WebObjects Developer 4.0 CD as an optional package. To install the Sybase client libraries, you must do a custom installation and explicitly specify that you want to install the package. To use

Enterprise Objects Framework with Oracle or Informix, you must purchase the appropriate client libraries from your database vendor.

Oracle

Phone: (800) 542-1170 or call your local sales representative

Ask for: Oracle 8 Client

The Oracle adaptor on NT requires the Oracle 8.0, 7.3, or 7.2 Client Library. It won't work with the 7.1 libraries.

On Windows NT, using the latest release of the Oracle client library (8.0) requires you to use SQL*Net v2, which requires a tnsnames.ora file. tnsnames.ora is a file that you put on client machines, generally in the directory Orant/Network/Admin. The file contains information needed to connect to a server over the network. Entries in tnsnames.ora are keyed off of a server ID alias, and they include information such as the server ID, the host machine name, and the network protocol used by the client library to resolve the server ID alias. An entry in tnsnames.ora might resemble the following:

```
myServerAlias = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=myMachine) (PORT=1521)) (CONNECT_DATA=(SID=eof)))
```

Oracle provides tools you can use to create tnsnames.ora files. Refer to your Oracle documentation for more information on tnsnames.ora files and the tools you can use to create them.

If you're using the 7.2 version of the Oracle client libraries on Windows NT, you can use either SQL*Net v1 or SQL*Net v2. To use SQL*Net v1, you should set your adaptor's connectionDictionary serverId entry to

```
"T:<host-machine>:<server-name>" .
```

Informix

Phone: (800) 331-1763 or call your local sales representative

Ask for: ESQL/C version 7.23.TC9 for Win32

Informix Notes

If you get the error "INFORMIXSERVER not in sqlhosts file (25596)" but can connect to your database server using the Informix **ilogin** program, you may need to run SetNet32 to update the environment variables used by Informix.

The Informix client libraries appear to have redundant sources of server information. They use the **sqlhosts** file (**\$INFORMIXDIR/etc/sqlhosts**) as well as a collection of environment variables managed by the Setnet32 program.

See your Informix documentation for more information on the **sqlhosts** file and the Setnet32 program.

Sybase

Phone: (800) 685-8225 or call your local sales representative

Ask for: OpenClient/C Version 11.1

Solaris Post-Installation Steps

After you've finished installing on Solaris, perform the following steps:

- Install the Java Native Threads Pack
- Rebuild the Executable WODefaultApp
- Obtain and Install Database Client Libraries

Install the Java Native Threads Pack

If you are running Solaris 2.5.1 (with patches 103566, 103600, and 103640), you must install the Java Native Threads Pack. You can download the Native Threads Pack from JavaSoft's Java Development Kit website.

Rebuild the Executable WODefaultApp

On Solaris, the Enterprise Objects Framework cannot automatically load your database's client library and its adaptor, as it can on other platforms. Because of this, you must rebuild the **WODefaultApp** executable, which is installed with WebObjects and is used to run purely scripted applications. If you don't rebuild this executable, any purely scripted applications you run with **WODefaultApp** won't be able to access a database.

If you answered “y” to all questions you were asked during installation, the **WODefaultApp** executable has already been rebuilt by the installation process. If you answered “n” to the question about building **WODefaultApp** or you have installed new client libraries afterwards, you should rebuild **WODefaultApp** before testing your installation.

To rebuild **WODefaultApp**, run the **RebuildWODefaultApp** script located in **/Developer/Examples/WebObjects/Source/WODefaultApp**.

Note: Each time you create a new project, you'll need to set it up so that it statically links the database's client library and adaptor. To do so, add the appropriate adaptor framework to the FRAMEWORKS makefile variable definition, and uncomment this line in the **Makefile.preamble**:

```
include $(MAKEFILEDIR)/pdo-eoadaptor-linking.make
```

Obtain and Install Database Client Libraries

To use Enterprise Objects Framework on Solaris, you must have the appropriate database client libraries.

Here's what you need:

Oracle

Phone: (800) 542-1170 or call your local sales representative

Ask for: 8.0 SQLNet V2 TCP/IP Client libraries

The Oracle adaptor on Solaris requires the Oracle 8.0 or 7.3 Client Library. The makefiles are configured for 7.3 or 8.0 (they default to 7.3); with some modification of the makefiles, you can also link with 7.2.

Informix

Phone: (800) 331-1763 or call your local sales representative

Ask for: ESQL/C Version 7.23.UC9

If you get the error "INFORMIXSERVER not in sqlhosts file (25596)" but can connect to your database server using the Informix ilogin program, you may need to run SetNet32 to update the environment variables used by Informix.

The Informix client libraries appear to have redundant sources of server information. They use the sqlhosts file (\$INFORMIXDIR/etc/sqlhosts) as well as a collection of environment variables managed by the Setnet32 program.

See your Informix documentation for more information on the sqlhosts file and the Setnet32 program.

Sybase

Phone: (800) 685-8225 or call your local sales representative

Ask for: OpenClient/C Version 11.1

Linking Against the Adaptor

On Solaris applications must explicitly link against the adaptor framework and the client libraries. New makefiles look for adaptor frameworks and automatically add in the right linker arguments. Simply add the adaptor framework to your project, and set the requisite environment variable specifying where the client libraries are installed. For Oracle set ORACLE_HOME and optionally ORACLE_REL. (The ORACLE_REL flag controls which set of libraries are used. It uses the Oracle 7.3 static link libraries by default, but you can also specify "8.0-static" or "7.3-dynamic.") For Sybase set SYBASE_HOME. For Informix set INFORMIX_HOME.

If you use dynamic libraries on Solaris, you need to set the LD_LIBRARY_PATH environment variable when running your application.

HP-UX Post-Installation Steps

After you've finished installing on HP-UX, perform the following steps:

- Rebuild the Executable **WODefaultApp**
- Change the UID for User nobody
- Obtain and Install Database Client Libraries
- Build the Examples

Rebuild the Executable **WODefaultApp**

On HP-UX, the Enterprise Objects Framework cannot automatically load your database's client library and its adaptor, as it can on other platforms. Because of this, you must rebuild the **WODefaultApp** executable, which is installed with WebObjects and is used to run purely scripted applications. If you don't rebuild this executable, any purely scripted applications you run with **WODefaultApp** won't be able to access the database.

If you answered “y” to all questions you were asked during installation, the **WODefaultApp** executable has already been rebuilt by the installation process. If you answered “n” to the question about building **WODefaultApp** or you have installed new client libraries afterwards, you should rebuild **WODefaultApp** before testing your installation.

To rebuild **WODefaultApp**, run the **RebuildWODefaultApp** script located in **/Developer/Examples/WebObjects/Source/WODefaultApp**.

Note: Each time you create a new project, you'll need to set it up so that it statically links the database's client library and adaptor. To do so, add the appropriate adaptor framework to the FRAMEWORKS makefile variable definition, and uncomment this line in the **Makefile.preamble**.

```
include $(MAKEFILEDIR)/pdo-eoadaptor-linking.make
```

Change the UID for User nobody

If you're using the Apache web server you'll need to change the UID of the user nobody, which is used to launch CGI processes. By default, the UID is -2, which causes setuid to complain about an invalid argument.

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

Obtain and Install Database Client Libraries

To use Enterprise Objects Framework on HP-UX, you must have the appropriate database client libraries.

Oracle

Phone: (800) 542-1170 or call your local sales representative

Ask for: 8.0 SQLNet V2 TCP/IP Client libraries

The Oracle adaptor on HP-UX requires the Oracle 8.0 or 7.3 Client Library. The makefiles are configured for 7.3 or 8.0 (they default to 7.3); with some modification of the makefiles, you can also link with 7.2.

Informix

Phone: (800) 331-1763 or call your local sales representative

Ask for: ESQL/C Version 7.23.UC6

If you get the error "INFORMIXSERVER not in sqlhosts file (25596)" but can connect to your database server using the Informix `ilogin` program, you may need to run `SetNet32` to update the environment variables used by Informix.

The Informix client libraries appear to have redundant sources of server information. They use the `sqlhosts` file (`$INFORMIXDIR/etc/sqlhosts`) as well as a collection of environment variables managed by the `Setnet32` program.

See your Informix documentation for more information on the `sqlhosts` file and the `Setnet32` program.

Sybase

Phone: (800) 685-8225 or call your local sales representative

Ask for: OpenClient/C Version 11.1

Linking Against the Adaptor

On HP-UX applications must explicitly link against the adaptor framework and the client libraries. New makefiles look for adaptor frameworks and automatically add in the right linker arguments. Simply add the adaptor framework to your project, and set the requisite environment variable specifying where the client libraries are installed. For Oracle set ORACLE_HOME and optionally ORACLE_REL. (The ORACLE_REL flag controls which set of libraries are used. It uses the Oracle 7.3 static link libraries by default, but you can also specify "8.0-static" or "7.3-dynamic.") For Sybase set SYBASE_HOME. For Informix set INFORMIX_HOME.

Build the Examples

The section *Verifying the Installation* describes how to verify that your installation is working properly by running the examples, some of which are compiled examples. On HP-UX, you must build the examples before you can test your system. If you want to use the WebObjects examples to test your system, build the examples found in **/Developer/Examples/WebObjects**.

Using Microsoft Access

Read this section if you want to use Enterprise Objects Framework to connect to a Microsoft Access database. The information in this section describes what you need to do before you can install the sample databases or create your own custom databases.

What You Need

Enterprise Objects Framework uses ODBC (a standard API developed by Microsoft for accessing database management systems) to interact

with Access databases, so you'll need an Access-specific ODBC driver if you don't have one already.

To determine if you've already got an ODBC driver installed, open the Control panel. If there isn't an ODBC option listed, then you don't have any ODBC drivers installed. The latest ODBC manager and drivers for Windows NT are available from Microsoft. See <http://www.microsoft.com/data/mdac15.htm>.

Some bugs in old versions of the ODBCJT32.DLL (used by the Enterprise Objects Framework to communicate with Microsoft Access) can cause problems with primary key generation in the ODBC Adaptor (the ODBC manager reports an invalid SQL statement when updating the EO_PK_TABLE). ODBCJT32.DLL version 3.40.2728 causes this problem, but the more recent ODBCJT32.DLL version 3.51.1029 works correctly. To verify that you don't have the older version of the ODBC drivers, open the ODBC Control Panel and click on the tab labeled "ODBC Drivers".

Creating Data Sources and Databases

After you've installed an ODBC driver for Access, you'll need to create databases and corresponding data sources for them. A data source simply stores the information needed to connect to your database. The easiest way to create an Access database and a corresponding data source from scratch is to create them both at the same time with the ODBC Data Source Administrator:

1. Choose the ODBC option in the Control Panel.

An ODBC Data Source Administrator panel opens.

2. Click the System DSN tab.

System data sources are accessible to all users, including NT services. It's important to create system data sources instead of user data sources, especially for use with WebObjects, because autolaunched applications aren't able to access user data sources.

3. Click Add.

The Create New Data Source panel opens.

4. Select the Microsoft Access Driver.
5. Click Finish.

The ODBC Microsoft Access 97 Setup panel opens.

6. Type a name for your data source in the Data Source Name field (Movies, for example).

Remember the name you provide because later you'll use it to login to your database.

7. In the Database section of the panel, click Create.

A New Database panel opens. You'll use this panel to create a new, empty database.

8. Type a name for your database in the Database Name field (Movies.mdb, for example).

9. Choose a directory where you want the database located.

10. Click OK.

A panel opens telling you that the database was successfully created.

11. In the ODBC Microsoft Access 97 Setup panel, click OK.

Your database and data source are now ready to use. If you want to populate the new database with one of the sample databases that come with WebObjects, you've completed the first step: setting up the database accounts. To find out what to do next, see "Setting Up the Sample Databases."

Setting Up the Sample Databases

WebObjects includes Enterprise Objects Framework database integration technology. If you've installed WebObjects Developer, several examples demonstrate Enterprise Objects Framework

programming techniques. In particular, if you work through the tutorials in the book *Getting Started With WebObjects*, you'll need two sample databases: Movies, which contains background information on all movies available from a store's distributor, and Rentals, which contains the inventory, customer list, and rental transaction records for the store. In addition, WebObjects ships with an example application named Movies, which uses a database of Movies, their ratings, and principal actors. The section *Verifying the Installation* uses Movies to verify that the installation is working properly.

Warning: None of the data in the sample databases is intended to be accurate.

All of the examples supplied with this release come pre-built, and are runnable directly from their distribution directories. Those that use the Enterprise Objects Framework run "out of the box" against the supplied OpenBaseLite databases. On Mac OS X Server and Windows NT systems, you can invoke the Setup Wizard to reconfigure the example application models to use another supported database, such as Oracle or Sybase, and to optionally load the database for you. The Setup Wizard copies the examples into a directory of your choosing before modifying them.

For WebObjects 4.0, the Setup Wizard doesn't completely perform all operations necessary to get the examples running. In particular, it doesn't convert the models in the BusinessLogicInheritance framework, and it doesn't load the inheritance model data into a database. However, it does work correctly on the BusinessLogic framework; thus, the simpler examples that don't use inheritance will work correctly with your database after you run the SetupWizard and perform a few additional manual steps.

Running the Setup Wizard

SetupWizard.app is located in **/System/Developer/Examples/EnterpriseObjects** (on Windows NT systems, **NEXT_ROOT\Developer\Examples\EnterpriseObjects**). After you invoke the wizard, you'll be prompted for a directory into which the examples should be installed. *This directory must already exist in order for the wizard to work properly.*

Warning: The wizard will delete any existing contents of the directory you specify.

After advancing to the next screen, follow the prompts through the rest of the wizard. Note that you'll need to select a database adaptor and login to the database twice: once for Movies, and once for Rentals. Finally, the wizard will ask you whether or not you want it to populate the selected databases for you.

After running the Setup Wizard, you'll need to perform a couple of additional steps. These steps are outlined in a separate online document, *ExampleGuide.rtf*, which is located in the same directory in which the Setup Wizard itself is located (**.../Examples/EnterpriseObjects**).

Verifying the Installation

At this point, you should have installed WebObjects as described in the *WebObjects Installation Guide* (included with the WebObjects 4.0 CD), perform the post-installation steps for your operating system as described in the first part of this document, and rebooted your computer.

After you have completed the installation and the post-installation, verify the installation by performing the following steps:

Note: These steps assume you've installed WebObjects Developer. The examples mentioned below are not installed with WebObjects Deployment. To verify a Deployment installation, you might complete these same steps running other applications.

1. Try to run a simple scripted application.

Open a command-shell window (on Windows NT, open a Bourne shell window) and enter the following commands:

```
> cd NEXT_ROOT/Developer/Examples/WebObjects/WebScript/  
HelloWorld  
  
> NEXT_ROOT/Library/WebObjects/Executables/  
WODefaultApp[.exe]
```

where:

NEXT_ROOT is the directory in which you installed WebObjects software (*NEXT_ROOT* isn't applicable on Mac OS X systems).

These commands should run the HelloWorld example application, launch a web browser, and enter HelloWorld's URL in the browser. If this doesn't work, go to the topic *Troubleshooting*.

After you have verified that HelloWorld runs, type Control-C to shut it down.

2. If scripted applications work, try running compiled applications. If your operating system has Java support (Mac OS X Server, Windows NT, or Solaris), enter the following commands:

```
> cd ../../Java/Movies/Movies.woa
> ./Movies[.exe]
```

If your operating system does not have Java support (HP-UX), open the Objective-C version:

```
> cd
../../ObjectiveC/HelloWorldCompiled/HelloWorldCompiled.woa
> HelloWorldCompiled
```

If you have trouble running compiled applications, go to the topic *Troubleshooting*.

After you have verified that Movies or HelloWorldCompiled runs, type Control-C to shut it down.

Troubleshooting

WebObjects operates on a number of hardware platforms, running various operating systems and supporting many different types of HTTP servers. Taking these variables together means that WebObjects finds itself in a large number of distinctly different environments, a situation that can lead to problems affecting the installation of WebObjects or the running of WebObjects applications.

This short guide will help you find solutions to the problems most commonly encountered. It's divided into these major sections:

- Checking the Installation
 - Windows NT
 - Solaris or HP-UX
 - Mac OS X Server
- Problems With Scripted Applications
- Problems With Compiled Applications

Start by reading the section for your operating system in “Checking the Installation.” If you don’t see the solution there, continue with “Problems With Scripted Applications” and then “Problems With Compiled Applications.”

Remember that the WebObjects release notes, available online (on your disk and on Apple’s web site), have the latest information about bugs and workarounds.

Checking the Installation

Windows NT

On Windows NT, these locations should contain the following files or directories:

- In your server’s *cgi-bin* directory:
 - **WebObjects.exe**: The WebObjects adaptor
 - **WebObjects**: This file is simply a copy of the WebObjects.exe file above minus the .exe extension. Some HTTP servers disallow the extension.
- Your server’s *document root* directory

- **WebObjects:** This directory should contain Documentation and Java subdirectories, plus a Frameworks subdirectory that contains directories for WOExtensions and DirectToWeb.
- *NEXT_ROOT* directory

This is the location where you installed the WebObjects software. Check for these files and directories:

- **Library\WebObjects\Executables:** Contains **WODefaultApp.exe**, the default application executable for scripted WebObjects applications
- **Library\Frameworks\WebObjects.framework:** WebObjects library of classes, plus header files (Developer installations only)
- **Library\Frameworks\WOExtensions.framework:** WebObjects Extensions framework, which contains extra dynamic elements and shared components
- **Library\Java:** The Java interfaces to WebObjects classes.
- **Library\WebObjects\Adaptors:** Contains WebObjects configuration files and adaptors.

Corrective actions:

If you are missing any of the files from your server's *cgi-bin* directory, you can copy them from **Library\WebObjects\Adaptors\CGI** to your HTTP server's *cgi-bin* directory.

If you are missing any of the contents of your server's *document root* directory copy them from where they were installed to the directory `<DocRoot>\WebObjects`. If the files were not installed, reinstall WebObjects.

If you are missing any of the contents of the **NEXT_ROOT** directory, reinstall WebObjects.

Solaris or HP-UX

On Solaris and HP-UX, these locations should contain the following files or directories:

- Your server's *cgi-bin* directory.
 - **WebObjects:** The WebObjects adaptor
- Your server's *document root* directory
 - **WebObjects:** This directory should contain Documentation and Java subdirectories, plus a Frameworks subdirectory that contains directories for WOExtensions and DirectToWeb.
- *NEXT_ROOT* directory

This is the location where you installed the WebObjects software. Check for these files and directories:

- **Library/WebObjects/Executables:** Contains **WODefaultApp**, the default application executable for scripted WebObjects applications
- **Library/Frameworks/WebObjects.framework:** WebObjects library of classes, plus header files (Developer installations only)
- **Library/Frameworks/WOExtensions.framework:** WebObjects Extensions framework, which contains extra dynamic elements and shared components
- **Library/Java:** The Java interfaces to WebObjects classes. (Solaris only)
- **Library/WebObjects/Adaptors:** Contains WebObjects configuration files and adaptors.

Corrective action:

If you are missing any of the files from your server's *cgi-bin* directory, you can copy them from **/Library/WebObjects/Adaptors/CGI** to your HTTP server's *cgi-bin* directory.

If you are missing any of the contents of your server's *document root* directory copy them from where they were installed to your server's *document root* directory. If the files were not installed, reinstall WebObjects.

If you are missing any of the contents of the **NEXT_ROOT** directory, reinstall WebObjects.

Mac OS X Server

On Mac OS X Server, these locations should contain the following files or directories:

- Your Apache server's CGI_Executables directory (**/Local/Library/WebServer/CGI-Executables**).
 - **WebObjects**: The WebObjects adaptor
- Your server's *document root* directory
 - **WebObjects**: This directory should contain Documentation and Java subdirectories, plus a Frameworks subdirectory that contains directories for WOExtensions and DirectToWeb.
- **/System/Library** directory

Check for these files and directories:

- **WebObjects/Executables**: Contains **WODefaultApp**, the default application executable for scripted WebObjects applications
- **Frameworks/WebObjects.framework**: WebObjects library of classes, plus header files (Developer installations only)
- **Frameworks/WOExtensions.framework**: WebObjects Extensions framework, which contains extra dynamic elements and shared components
- **Java**: The Java interfaces to WebObjects classes.
- **WebObjects/Adaptors**: Contains WebObjects configuration files and adaptors.

Corrective action:

If you are missing any of the files from your server's *cgi-bin* directory, check to see that you have moved them from **/Local/Library/WebServer/CGI-Executables** to your HTTP server's CGI-Executables directory.

Likewise, if you are missing any of the directories from your server's *document root* directory, check to see that you have moved them from **/Local/Library/WebServer/Documents** to your HTTP server's *document root* directory.

If you are missing any of the contents of your **/System/Library** directory, reinstall WebObjects.

Problems With Scripted Applications

Scripted example applications (HelloWorld, TimeOff, etc.) are the simplest ones and don't contain compiled code.

Problem

The web browser does not launch or launches the incorrect URL

Checklist

1. Check the debugging statements printed in the command-shell window.

When you launch a WebObjects application from the command line, the application computes its own URL, launches the web browser, and enters the URL in the browser. It prints messages about the values it computes to standard output.

Check the standard output (the command-shell window) for these messages (among others):

```
The applicationPath:  
/System/Developer/Examples/WebObjects/WebScript/HelloWorld  
The applicationBaseURL: /WebObjects/HelloWorld  
Opening application's URL in Browser: url
```

Corrective action:

If you see these messages but your web browser doesn't launch, you might not have a browser installed on your system, or WebObjects cannot find the browser. This is always true on Solaris and HP-UX. If the URL looks correct (as described below), open your browser and type that URL into it.

If you see a message that says "No Adaptor URL in WebServerConfig.plist," either the **WebServerConfig.plist** file is missing, or the **WOAdaptorURL** key is missing from it. The file should look something like this:

```
{
    DocumentRoot = "/Apple/Library/WebServer/Documents" ;
    WOADaptorURL = "http://localhost/cgi-bin/WebObjects" ;
}
```

If **WOAdaptorURL** is missing, the web browser does not launch when you launch a WebObjects application. You can enter **WOAdaptorURL** or you can type the URL in the browser and connect to the running application that way.

This base URL value of **WOAdaptorURL** is of the form:

```
http://localhost/cgi-bin/WebObjects
```

cgi-bin is the name of your HTTP server's cgi-bin directory. You specify this name when you configure your HTTP server. The *cgi-bin* directory name is often **cgi-bin**, but it may have a different name. For example, the Microsoft Internet Information Server uses the name **Scripts**.

WebObjects is the name of the WebObjects CGI adaptor as you see it in your HTTP server's cgi-bin directory. Usually, the name is WebObjects. If you're using Windows NT, the adaptor name might be **WebObjects.exe** (however, some older Netscape servers don't use the **.exe** extension.)

If the base URL's cgi-bin and WebObjects adaptor names look correct, consider the **localhost** value. On most sites, **localhost** accesses the server on the local host. However, some sites require

a domain name as well (**http://localhost.apple.com**). If your HTTP server isn't running on your local machine, use the host name of the machine running the server in place of “localhost” in the URL above, and make sure a WebObjects adaptor is installed on that machine.

Problem

A simple scripted application won't run properly.

Checklist

1. Try using direct-connect to access your WebObjects application.
2. Check that you can load a static page.

Corrective action:

If your browser displays a message saying that it was unable to connect or that the connection was refused, your HTTP server is probably not running. Check that your server is running. Otherwise, see *Checking the Installation* for information on how to fix your installation of WebObjects.

3. Check that the WebObjects adaptor is functioning.

Check that the WebObjects adaptor is installed correctly and can run. Use your browser to open this URL (which specifies the WebObjects adaptor, but fails to specify an application name):

```
http://localhost/cgi-bin/WebObjects
```

(You may need to replace “localhost” with the name of the host running your HTTP server. You may also need to replace “cgi-bin” with the actual name of the directory that contains scripts and CGI programs on your server.) If the WebObjects adaptor is installed correctly, it returns the following error message:

```
Invalid application name
```

If the adaptor is installed incorrectly or can't run, the browser will instead display a message indicating that the requested object cannot be located. The message may look like this:

```
404 Not Found
The requested URL /cgi-bin/WebObjects was not found on
this server.
```

Corrective action:

Make sure you've supplied the right names in the URL for the host (“localhost” in the example above) and for the cgi-bin directory (sometimes named “Scripts” or “cgiPrograms” rather than “cgi-bin”). Otherwise, see *Checking the Installation* for information on how to fix your installation of WebObjects.

Problems With Compiled Applications

Running compiled applications exercises more features of the WebObjects framework and development environment. Before attempting to troubleshoot a problem with running a compiled application, make sure you can run a simple scripted application.

Problem

A simple compiled application won't run properly.

Checklist

1. Make sure the executable has been built.

On the HP-UX platform, examples are not installed built. You must build them yourself before you can run them. To compile an example, **cd** to the example's project directory and type **make**. For example:

```
> cd /Apple/Developer.Examples/WebObjects/ObjectiveC/
HelloWorldCompiled
> make
```

Problem

The Movies application won't run.

Checklist

1. Make sure the application was correctly installed and compiled.

The Movies application must be compiled before you can run it. In addition, you must create the Movies database (scripts are provided) and install the database model file that is compatible with your database server as described in “Setting Up the Sample Databases.”

Check the Movies directory for a directory named **Movies.woa**. This is the WebObjects application wrapper. Check the wrapper for an executable file. If the wrapper or the executable doesn't exist, build the Movies application.

On Solaris and HP-UX, you need to build Movies with the correct client libraries and adaptor. Before you build, add the appropriate adaptor framework to the FRAMEWORKS makefile variable. Then uncomment the following line in the **Makefile.preamble** to link the appropriate client libraries:

```
include $(MAKEFILEDIR)/pdo-eoadaptor-linking.make
```

If Movies compiles and runs but can't access data about the various movies, it's probably because the application can't communicate with the database server.

Problem

A WebObjects application won't connect to the database server.

Checklist

1. Check that your database server itself is operating correctly.

Check that the client libraries for your database server are correctly installed on your machine. If so, you can, for example, use the tools supplied with the database server (`isql` for Sybase, `sqlplus` for Oracle, and `dbaccess` for Informix) to test that you can connect to the server and execute simple SQL commands.

2. Make sure that the database model file is accessible to your application

The model file should be in the **Resources** directory under the application's **.woa** directory. Taking the Movies application for example, the directory structure would look like this:

```
Movies.woa/  
  Movies (the executable file)  
  Resources/Movies.eomodeld (the model file)
```

