



# BEA WebLogic Server<sup>™</sup> and BEA WebLogic Express<sup>™</sup>

## Release Notes for BEA WebLogic Server 6.1

BEA WebLogic Server Version 6.1  
Document Date: July 14, 2004

## Copyright

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Collaborate, BEA WebLogic Commerce Server, BEA WebLogic E-Business Platform, BEA WebLogic Enterprise, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Server, E-Business Control Center, How Business Becomes E-Business, Liquid Data, Operating System for the Internet, and Portal FrameWork are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

### BEA WebLogic Server 6.1 Release Notes

<b>Part Number</b>	<b>Document Date</b>	<b>Software Version</b>
860-001003-010	July 14, 2004	BEA WebLogic Server Version 6.1

---

# Contents

## 1. WebLogic Server 6.1 Features and Changes

WebLogic Server 6.1 Service Pack 7 .....	1-2
Oracle 10g Thin Driver Bundled with WebLogic Server .....	1-2
WebLogic Server 6.1 Service Pack 6 .....	1-2
Changes in EJB .....	1-3
Changes in System Administration .....	1-3
WebLogic Server 6.1 Service Pack 5 .....	1-3
WebLogic Server 6.1 Service Pack 4 .....	1-3
Changes in System Administration .....	1-4
Change in Behavior of -delay Argument for beasvc.exe .....	1-4
New Parameter for Forced Application Update.....	1-4
Setting Default Encoding for JVM .....	1-5
Changes in JDBC .....	1-6
WebLogic Server 6.1 Service Pack 3 .....	1-7
Changes in System Administration .....	1-7
Strict Enforcement of ACLs .....	1-7
Changes in WebLogic jDriver Support .....	1-7
Oracle 9.0.1 Thin Driver Certification .....	1-8
Changes in JMS.....	1-8
WebLogic-Tuxedo Interoperability.....	1-9
WebLogic Server 6.1 Service Pack 2 .....	1-10
Compatibility Issues .....	1-10
Start Script Switch for WebLogic Server 6.1 SP02 Clients.....	1-10
SAX Parser and Encoding Error Detection.....	1-10
Certifications .....	1-11
Changes to Proxy Servlets.....	1-11

---

New Security Classes .....	1-12
Servlet and JSP Encoding Changes .....	1-12
WebLogic 6.1 SP02 and 5.1 Interoperability .....	1-13
WebLogic Server 6.1 Service Pack 1 .....	1-13
WebLogic Server 6.1 (G.A. Release) .....	1-13
Notes for WebLogic APIs .....	1-14
Applets .....	1-15
Beanshell Code .....	1-15
Clustering .....	1-15
Deployment in WebLogic Server 6.1 .....	1-15
Enterprise Java Beans (EJBs) .....	1-16
Hypertext Transfer Protocol (HTTP) 1.1 .....	1-17
J2EE Connector .....	1-18
Java Database Connectivity (JDBC) .....	1-19
JavaMail .....	1-19
Java Transaction API (JTA) .....	1-19
JMS .....	1-19
JSP .....	1-21
Security .....	1-22
Servlets and Web Applications .....	1-22
SNMP .....	1-24
System Administration .....	1-24
WebLogic Tuxedo Connector .....	1-26
XML .....	1-27
Web Services .....	1-28
Deprecated Features and APIs .....	1-28
Documentation and Examples .....	1-29
WebLogic Server Tour .....	1-30
J2EE and Standards Topics .....	1-30
Java Development Kit .....	1-30
WebLogic Server 6.1 with J2EE 1.2 Plus Additional J2EE 1.3 Features 1-30	
WebLogic Server 6.1 with J2EE 1.2 Certification .....	1-34
Product CD Installers for J2EE 1.2 and 1.3 .....	1-34
Standards Support .....	1-34

---

## 2. Upgrading WebLogic Server 5.1 to Version 6.1

Upgrading Your WebLogic Server Configuration: Main Steps.....	2-2
Converting the weblogic.properties File .....	2-2
Classloading in WebLogic Server 6.x .....	2-3
Modifying Startup Scripts .....	2-4
Upgrading Oracle .....	2-4
Upgrading JMS.....	2-5
JSP .....	2-13
Upgrading JVM.....	2-13

## 3. Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

The weblogic.properties File and .xml Files .....	3-2
WebLogic Server 6.x Application Types .....	3-2
Migration Main Steps .....	3-3
Converting the weblogic.properties File to .xml files .....	3-3
weblogic.properties Mapping Table .....	3-5
Migrating Web Applications .....	3-24
Web Applications Directory Structure .....	3-25
XML Deployment Descriptors .....	3-26
WAR Files .....	3-27
Deploying Web Applications .....	3-27
Session Migration.....	3-28
JavaServer Pages (JSPs) and Servlets .....	3-28
Migrating a Simple Servlet from WebLogic Server 5.1 to WebLogic Server 6.1 .....	3-29
Migrating and Converting Enterprise JavaBeans Applications.....	3-31
EJB Migration Considerations .....	3-31
EJB Migration Recommendations .....	3-32
Steps for Migrating a 1.0 EJB from WebLogic Server 4.5.x to WebLogic Server 6.1 .....	3-34
Steps for Migrating a 1.1 EJB from WebLogic Server 5.1 to WebLogic Server 6.1 .....	3-35
Steps for Converting an EJB 1.1 to an EJB 2.0.....	3-36
Migrating EJBs from other J2EE Application Servers .....	3-36
Creating an Enterprise Application .....	3-37

---

Understanding J2EE Client Applications .....	3-38
Migrating JMS Applications .....	3-39
Additional Migration and Deployment Considerations.....	3-40
Standalone HTML and JSPs.....	3-40
Applications and Managed Servers .....	3-41
Java Transaction API (JTA) .....	3-41
Java Database Connectivity (JDBC) .....	3-41
RMI.....	3-42
FileServlet.....	3-43
Internationalization (I18N).....	3-43
Security .....	3-44
WAP Applications.....	3-45
Session Migration.....	3-45
Web Components .....	3-46
XML 6.0->6.1 Parser.....	3-47
Deprecated APIs and Features.....	3-47
Removed APIs and Features .....	3-48

## **4. Migrating WebLogic Server 6.0 Applications to WebLogic Server 6.1**

Migrating an Application from 6.0 to 6.1.....	4-2
JMS .....	4-3
EJB 2.0.....	4-4
Upgrading EJB 2.0 Session Beans and BMP Entity Beans.....	4-4
Upgrading EJB 2.0 Message-Driven Beans (MDB) .....	4-5
Upgrading EJB 2.0 CMP Entity Beans .....	4-5
JMX .....	4-6
Apache XML Parser .....	4-6
Xalan XML Parser .....	4-6
Web Applications .....	4-7
Security .....	4-7
config\applications Directory .....	4-8
WebLogic Server Clusters on Solaris.....	4-8
Thread Pool Size.....	4-8

---

## 5. Notes and Problems

Connector Known Problems.....	5-3
Console Known Problems .....	5-3
Core Known Problems .....	5-4
EJB Known Problems.....	5-6
Examples and Pet Store Demo Known Problems .....	5-10
Installer Known Problems .....	5-15
JCA Known Problems .....	5-17
JDBC and jDrivers Known Problems.....	5-17
JMS Known Problems .....	5-20
JSP Known Problems .....	5-20
JVM Known Problems .....	5-22
Plug-Ins Known Problems.....	5-24
RMI Known Problems.....	5-25
RMI-IIOP Known Problems.....	5-25
Security Known Problems.....	5-26
Server Known Problems.....	5-28
Servlet and Web Application Known Problems.....	5-31
System Administration Known Problems .....	5-34
Tools Known Problems .....	5-38
Web Services Known Problems .....	5-38
WebLogic Tuxedo Connector Known Problems .....	5-41
XML Known Problems .....	5-43
ZAC Known Problems .....	5-44

## 6. Resolved Problems

WebLogic Server 6.1 Service Pack 7 Solutions.....	6-1
Administration Console.....	6-3
Classloader .....	6-6
Cluster .....	6-6
Core .....	6-8
Deployment .....	6-12
EJB .....	6-12
Internationalization.....	6-15
JDriver .....	6-15

---

JDBC .....	6-17
JMS.....	6-19
JNDI .....	6-21
JSPs and Servlets.....	6-21
JVM.....	6-25
Operations, Administration and Management.....	6-25
Plug-Ins.....	6-27
RMI-IIOP .....	6-35
RMI.....	6-35
Samples.....	6-36
Security.....	6-36
SNMP .....	6-40
WLEC.....	6-40
WebLogic Tuxedo Connector .....	6-40
WebLogic Server 6.1 Service Pack 6 Solutions .....	6-41
Classloader .....	6-43
Cluster.....	6-44
Connector .....	6-48
Console .....	6-49
Core .....	6-51
Deployment .....	6-63
EJB .....	6-64
JDBC .....	6-76
jDriver.....	6-82
JMS.....	6-83
JNDI .....	6-86
JSP .....	6-88
JTA .....	6-94
Node Manager .....	6-98
OA&M (Operations, Administration, and Management).....	6-98
Plug-ins.....	6-107
RMI.....	6-114
RMI/IIOP.....	6-115
Security.....	6-117
Servlets .....	6-122

---

SNMP .....	6-132
Web Services.....	6-132
WTC-ATMI .....	6-134
XML.....	6-136
WebLogic Server 6.1 Service Pack 5 Solutions.....	6-136
Cluster .....	6-138
Connector .....	6-141
Console.....	6-142
Core .....	6-146
Deployment .....	6-160
EJB .....	6-162
Installer.....	6-173
JDBC .....	6-173
jDriver .....	6-179
JMS.....	6-182
JNDI .....	6-190
JSP .....	6-192
JTA .....	6-194
Miscellaneous.....	6-198
Plug-ins.....	6-198
RMI-IIOP .....	6-207
RMI .....	6-208
Security.....	6-208
Servlets .....	6-210
WLS-Tour/Examples .....	6-224
Web Services.....	6-226
WebLogic Tuxedo.....	6-229
XML.....	6-230
WebLogic Server 6.1 Service Pack 4 Solutions.....	6-230
Classloader .....	6-232
Cluster .....	6-232
Console.....	6-234
EJB .....	6-235
Examples .....	6-237
JDBC and jDriver.....	6-237

---

JMS.....	6-239
Miscellaneous .....	6-241
Plug-ins.....	6-249
Servlets, JSPs, and Web Applications.....	6-251
Security.....	6-257
System Administration .....	6-259
Web Services.....	6-261
WebLogic Tuxedo .....	6-261
XML .....	6-262
WebLogic Server 6.1 Service Pack 3 Solutions .....	6-262
Classloader .....	6-264
Cluster.....	6-264
Console .....	6-266
Core .....	6-268
EJB .....	6-268
Examples .....	6-273
JDBC .....	6-273
jDriver.....	6-276
JMS.....	6-277
JTA .....	6-278
Miscellaneous .....	6-278
Plug-Ins.....	6-285
RMI over IIOP.....	6-288
Security.....	6-288
Servlets, JSPs and Web Applications.....	6-291
System Administration .....	6-298
Web Services.....	6-301
WebLogic Tuxedo Connector .....	6-303
XML .....	6-303
Third-Party JAR Files in WebLogic Server 6.1 Service Pack 3 .....	6-305
WebLogic Server 6.1 Service Pack 2 Solutions .....	6-306
EJB .....	6-307
Examples .....	6-309
JDBC .....	6-309
jDriver.....	6-311

---

JMS.....	6-311
Miscellaneous.....	6-311
Plug-Ins .....	6-317
RMI over IIOP .....	6-317
Security.....	6-317
Servlets, JSP and Web Applications .....	6-319
System Administration.....	6-323
Web Services.....	6-325
WebLogic Tuxedo Connector .....	6-326
XML.....	6-326
WebLogic Server 6.1 Service Pack 1 Solutions.....	6-327
Deployment Descriptor Editor .....	6-328
EJB .....	6-330
Examples .....	6-334
JDBC .....	6-336
JMS.....	6-338
JTA.....	6-339
Miscellaneous.....	6-339
Plug-ins.....	6-344
RMI over IIOP .....	6-345
Servlets and JSP .....	6-345
System Administration.....	6-349
Web Services.....	6-352
XML.....	6-352
WebLogic Server 6.1 Release Solutions .....	6-353
Deployment Descriptor Editor .....	6-355
EJB .....	6-356
Examples .....	6-356
Internationalization.....	6-356
JCA.....	6-358
JDBC .....	6-358
JMS.....	6-360
Miscellaneous.....	6-364
Plug-ins.....	6-368
Servlets and JSP .....	6-370

---

Web Services .....	6-373
XML .....	6-373

---

# About This Document

This document introduces the current release of BEA WebLogic Server. It contains important information on new features, supported specifications, and known problems.

The document is organized as follows:

- Chapter 1, “WebLogic Server 6.1 Features and Changes,” is an overview of WebLogic Server 6.1. It contains important information concerning the current release of WebLogic Server and any accompanying Service Packs.
- Chapter 2, “Upgrading WebLogic Server 5.1 to Version 6.1,” contains information for users who are moving to WebLogic Server 6.1 from an earlier version of WebLogic Server.
- Chapter 3, “Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x,” is an overview of moving your application from an older version of WebLogic Server to WebLogic Server 6.0 or 6.1.
- Chapter 4, “Migrating WebLogic Server 6.0 Applications to WebLogic Server 6.1,” is an overview of moving your applications from WebLogic Server 6.0 to 6.1.
- Chapter 5, “Notes and Problems,” is a list of known problems in WebLogic Server 6.1.
- Chapter 6, “Resolved Problems,” contains information on bug fixes for WebLogic Server 6.1.

---

# Audience

This document is written for all users of WebLogic Server 6.1.

## e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

## How to Print the Document

You can print a copy of this document from a Web browser, one main topic at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the WebLogic Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at <http://www.adobe.com>.

## Contact Us!

Your feedback on BEA documentation is important to us. Send us e-mail at [docsupport@bea.com](mailto:docsupport@bea.com) if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the documentation.

---

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at <http://www.bea.com>. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

## Documentation Conventions

The following documentation conventions are used throughout this document.

<b>Convention</b>	<b>Usage</b>
Ctrl+Tab	Keys you press simultaneously.
<i>italics</i>	Emphasis and book titles.

Convention	Usage
monospace text	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that you enter from the keyboard. <i>Examples:</i> import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float
<i>monospace italic text</i>	Variables in code. <i>Example:</i> String <i>CustomerName</i> ;
UPPERCASE TEXT	Device names, environment variables, and logical operators. <i>Examples:</i> LPT1 BEA_HOME OR
{ }	A set of choices in a syntax line.
[ ]	Optional items in a syntax line. <i>Example:</i> java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]
	Separates mutually exclusive choices in a syntax line. <i>Example:</i> java weblogic.deploy [list deploy undeploy update] password {application} {source}
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> <li>■ An argument can be repeated several times in the command line.</li> <li>■ The statement omits additional optional arguments.</li> <li>■ You can enter additional parameters, values, or other information</li> </ul>

---

<b>Convention</b>	<b>Usage</b>
-------------------	--------------

---

- |   |  |
|---|--|
| . | Indicates the omission of items from a code example or from a syntax line. |
| . |  |
| . |  |
-



# 1 WebLogic Server 6.1 Features and Changes

The following sections describe changes and features in WebLogic Server 6.1.

The current release is WebLogic Server 6.1 Service Pack 6. Each Service Pack contains the changes and corrections introduced in prior Service Packs.

The following sections highlight the key changes in each WebLogic Server 6.1 Service Pack.

- “WebLogic Server 6.1 Service Pack 7” on page 1-2
- “WebLogic Server 6.1 Service Pack 6” on page 1-2
- “WebLogic Server 6.1 Service Pack 5” on page 1-3
- “WebLogic Server 6.1 Service Pack 4” on page 1-3
- “WebLogic Server 6.1 Service Pack 3” on page 1-7
- “WebLogic Server 6.1 Service Pack 2” on page 1-10
- “WebLogic Server 6.1 Service Pack 1” on page 1-13
- “WebLogic Server 6.1 (G.A. Release)” on page 1-13

For information about current known problems in WebLogic Server 6.1, see Chapter 5, “Notes and Problems.”

## WebLogic Server 6.1 Service Pack 7

These sections describe key changes in WebLogic Server 6.1 SP07.

- “WebLogic Server 6.1 Service Pack 7 Solutions” on page 6-1
- “Oracle 10g Thin Driver Bundled with WebLogic Server” on page 1-2

### Oracle 10g Thin Driver Bundled with WebLogic Server

In WebLogic Server 6.1 Service Pack 7, the Oracle 10g Thin driver is installed with WebLogic Server in the `WL_HOME\lib\oracle\10g` folder. BEA provides the 10g driver so that you can optionally use it with WebLogic Server. However, the Oracle 9.2.0 Thin driver remains the default version of the driver. To use the Oracle10g Thin driver, you must add the driver classes to your `CLASSPATH` in front of `weblogic.jar`. See “[Using the Oracle 10g Thin Driver](#)” in *Programming WebLogic JDBC* for more information.

**Note:** Oracle removed some methods and classes from the 10g Thin driver that were in previous versions of the driver. If you use any of the extension methods that were removed, you will see errors in your application.

## WebLogic Server 6.1 Service Pack 6

These sections describe bug fixes and key changes in WebLogic Server 6.1 SP06.

- “WebLogic Server 6.1 Service Pack 6 Solutions” on page 6-41
- “Changes in EJB” on page 1-3
- “Changes in System Administration” on page 1-3

## Changes in EJB

A field optimization was implemented for EJB 1.1 CMP beans, so that fields that are updated but whose values are not changed are not written to back to the database. The optimization applies to primitive and immutable fields only.

## Changes in System Administration

You must provide user credentials for any `weblogic.Admin` command that connects to a WebLogic Server instance. A new command, `STOREUSERCONFIG`, encrypts and stores user credentials, so that you do not have to pass unencrypted credentials on the command line or in scripts. For more information, see [STOREUSERCONFIG](#) in *WebLogic Server Administration Guide*.

# WebLogic Server 6.1 Service Pack 5

These sections describe bug fixes and key changes in WebLogic Server 6.1 SP05.

- “WebLogic Server 6.1 Service Pack 5 Solutions” on page 6-136

# WebLogic Server 6.1 Service Pack 4

These sections describe bug fixes and key changes in WebLogic Server 6.1 SP04:

- “WebLogic Server 6.1 Service Pack 4 Solutions” on page 6-230
- “Changes in System Administration” on page 1-4
- “Changes in JDBC” on page 1-6.

## Changes in System Administration

This section lists key system administration changes in WebLogic Server 6.1 SP04.

### Change in Behavior of `-delay` Argument for `beasvc.exe`

The behavior and usage of the `-delay` argument for `beasvc.exe`, the WebLogic Server Windows service program, is changed in WebLogic Server 6.1 SP04.

Previously, the `-delay` argument was applied when invoking `beasvc.exe` for Managed Server. When the Windows Service Control Manager (SCM) started a Managed Server as a service, it waited for the period (in milliseconds) specified with the `-delay` argument, set the service status to `STARTED`, and started the service.

In WebLogic Server 6.1 SP04, the `-delay` argument is applied at the Administration Server level, rather than for each Managed Server. Use the `-delay` argument when invoking `beasvc.exe` for the Administration Server for the domain. The argument affects all services that depend on the Administration Servers, including Managed Servers in the domain.

There are also changes in the effect that `-delay` has on service status and startup. In WebLogic Server 6.1 SP04, the Windows Service Control Manager (SCM) starts the Managed Server immediately, sets the service status to `SERVER_START_PENDING`, and after the specified delay has elapsed, sets the service status to `STARTED`.

### New Parameter for Forced Application Update

A new startup option for Administration Servers, `-Dweblogic.management.forceApplicationCopy`, forces Managed Servers to obtain the latest version of deployed applications at startup. The effect of the startup option is described in “Forcing Application Update at Startup”. “Default Application Update Process” describes how Managed Servers obtain updates when `-Dweblogic.management.forceApplicationCopy` is false, as it is by default.

#### Forcing Application Update at Startup

The parameter `-Dweblogic.management.forceApplicationCopy` can be specified as a startup option for an Administration Server, either on the command line or in a startup script. If an Administration Server is started with

-Dweblogic.management.forceApplicationCopy set to true, when a Managed Server in the domain starts up, the applications deployed to that Managed Server are copied from the Administration Server to the Managed Server.

## Default Application Update Process

If an Administration Server is not started with

-Dweblogic.management.forceApplicationCopy set to true, an application is only copied to a Managed Server at startup if both these conditions are true:

- the application has been modified and redeployed
- the Managed Server does not have the latest version of the application.

The Administration Server maintains a `StagedTargets` list that specifies which Managed Servers in a domain have the latest version of an application. At startup, a Managed Server queries its Administration Server to determine if it (the Managed Server) has the most recent version of the application. If an updated version of the application is available, it is copied to the Managed Server, and the Administration Server adds that Managed Server to the `StagedTargets` list.

When an Administration Server goes down, all Managed Servers are removed from the `StagedTargets`. When the Administration Server is rebooted, each Managed Server in the domain copies its deployed applications from the Administration Server. This assures that, in the event that an application has been updated while the Administration Server was down, all Managed Servers have the same version of the application.

## Setting Default Encoding for JVM

Customers running WLS 6.1 SP04 and Sun JVM 1.3.1\_06 attempted and failed to set the default encoding for the VM using the `-Dfile.encoding` attribute.

Since `file.encoding` is a read-only property this property cannot be changed, either on the command line or via the methods in the `java.lang.System` class.

Please see <<http://developer.java.sun.com/developer/bugParade/bugs/4163515.html>> and the related bugs for more details.

The preferred way to change the default encoding used by the VM and the runtime system is to change the locale of the underlying platform before starting your Java program.

## Changes in JDBC

With the release of WebLogic Server 6.1 SP04, BEA certified the Oracle 9.2.0 Thin Driver for use with WebLogic Server. The software distribution for WebLogic Server 6.1 SP04 includes the following new JDBC drivers:

- WebLogic jDriver for Oracle 9.2.0, in addition to versions for Oracle 8.1.7 and 9.0.1.
- Oracle 9.2.0 Thin Driver, which replaces the 8.1.7 version that shipped with previous releases of WebLogic Server 6.1.

For more information about using WebLogic jDriver for Oracle, see [Installing and Using WebLogic jDriver for Oracle](#). For more information about using the Oracle Thin Driver with WebLogic Server, see [Using Third-Party Drivers with WebLogic Server](#).

In the Oracle Thin Driver version 9.2.0, Oracle removed some extension methods that were previously available in the driver. These methods are no longer supported and are not included in WebLogic Server 6.1 SP04:

Class:

```
weblogic.jdbc.oracle.OracleConnection
```

Method:

```
isCompatibleTo816()
```

Class:

```
weblogic.jdbc.oracle.OracleStatement
```

Method:

```
getWaitOption()  
setWaitOption(int i)  
setAutoRollback(int i)  
getAutoRollback()
```

Class:

```
weblogic.jdbc.oracle.OracleResultSet
```

Method:

```
getCURSOR(String s)
```

# WebLogic Server 6.1 Service Pack 3

These sections describe bug fixes and key changes in WebLogic Server 6.1 SP03:

- “WebLogic Server 6.1 Service Pack 3 Solutions” on page 6-262
- “Changes in System Administration” on page 1-7
- “Changes in WebLogic jDriver Support” on page 1-7
- “Oracle 9.0.1 Thin Driver Certification” on page 1-8
- “Changes in JMS” on page 1-8

## Changes in System Administration

This section lists key system administration changes in WebLogic Server 6.1 SP03.

### Strict Enforcement of ACLs

In WebLogic Server 6.1 SP03, an error in the implementation of ACLs for JDBC was fixed, which makes ACLs very strict. If you define an ACL for connection pools, access is restricted to *exactly* what is defined in the ACL. For more information, see [Permissions](#) at

<http://e-docs.bea.com/wls/docs61/jdbc/programming.html#programming006>.

## Changes in WebLogic jDriver Support

The following list details recent changes to supported versions of Oracle used with WebLogic jDriver for Oracle:

- As of the release of WebLogic Server 6.1 SP03, BEA no longer supports WebLogic jDriver for Oracle with Oracle 8.1.6. If you are using WebLogic jDriver for Oracle to connect to an Oracle 8.1.6 database, you must upgrade to

Oracle version 8.1.7 or 9.0.1, which includes the following steps on the machines on which WebLogic Server is running:

- Upgrade the Oracle client installation to the 8.1.7 or 9.0.1 version.
- Update the `ORACLE_HOME` environment variable to point to the updated version of the Oracle client.
- Modify your `PATH` (on Windows, varies on UNIX) to include the path to the new Oracle client and to the appropriate version of the WebLogic `jdbc` driver for Oracle.

For more information, see [Setting Up the Environment for Using WebLogic `jdbc` Driver for Oracle](http://e-docs.bea.com/wls/docs61/oracle/install_jdbc.html#environment) in *Installing and Using WebLogic `jdbc` Driver for Oracle* at [http://e-docs.bea.com/wls/docs61/oracle/install\\_jdbc.html#environment](http://e-docs.bea.com/wls/docs61/oracle/install_jdbc.html#environment).

## Oracle 9.0.1 Thin Driver Certification

For the release of WebLogic Server 6.1 SP03, BEA certified the Oracle 9.0.1 Thin Driver for use with WebLogic Server. For more information about using the Oracle Thin Driver with WebLogic Server, see [Using Third-Party Drivers with WebLogic Server](http://e-docs.bea.com/wls/docs61/jdbc/thirdparty.html) at <http://e-docs.bea.com/wls/docs61/jdbc/thirdparty.html>.

## Changes in JMS

WebLogic Server 6.1 SP03 includes the WebLogic Messaging Bridge.

A messaging bridge is responsible for transferring messages between two messaging providers. The WebLogic Messaging Bridge allows you to configure a store-and-forward mechanism between any two messaging products—including separate implementations of WebLogic JMS.

For more information, see [Using the WebLogic Messaging Bridge](#) in the *Administration Guide*.

WebLogic Server 6.1 SP03 includes a new JMS message paging feature.

The WebLogic JMS Message Paging feature can free up valuable virtual memory during peak message load periods by swapping out messages from virtual memory to persistent storage when message loads reach a specified threshold. From a performance perspective, this feature can greatly benefit WebLogic Server implementations with the large message spaces that are required by today's enterprise applications.

Two metrics are used to determine when to start and stop paging: bytes paging and messages paging. Each metric is the basis of a single paging mode, which you can enable and disable individually, or use simultaneously, on JMS servers and/or destinations (topics and queues).

Paging is configured through the Administration Console. Using the paging attributes on the JMS Server node, you can specify a paging store for the server, enable bytes and/or messages paging, and configure the bytes/messages high and low thresholds to start and stop paging. Similarly, using the paging attributes on the Destinations node, you can configure bytes/messages paging for all topics and queues configured for the JMS server. The destinations use the paging store that is configured for the JMS server. Also, if you use JMS templates to configure multiple destinations, you can use the paging attributes on the Templates node to quickly configure message paging on all your destinations.

For detailed instructions on configuring message paging, see the section called [“Tuning JMS”](#) in the *WebLogic Server Administration Guide* at <http://e-docs.bea.com/wls/docs61/adminguide/jms.html>.

## WebLogic-Tuxedo Interoperability

WebLogic Server 6.1 SP3 interoperates with C++ clients using the Tuxedo 8.0 C++ Client ORB. Tuxedo release 8.0 RP 56 and above is required. WebLogic Server users should contact their BEA Service Representative for information on how to obtain the Tuxedo C++ Client ORB. For more information on how WebLogic Server interoperates with a Tuxedo C++ Client ORB, see [RMI-IIOP with Tuxedo and Tuxedo Clients](#).

# WebLogic Server 6.1 Service Pack 2

These sections describe bug fixes and key changes in WebLogic Server 6.1 SP02:

- “WebLogic Server 6.1 Service Pack 2 Solutions” on page 6-306
- “Compatibility Issues” on page 1-10
- “Certifications” on page 1-11
- “Changes to Proxy Servlets” on page 1-11
- “New Security Classes” on page 1-12
- “Servlet and JSP Encoding Changes” on page 1-12
- “WebLogic 6.1 SP02 and 5.1 Interoperability” on page 1-13

## Compatibility Issues

This section describes compatibility issues in WebLogic Server 6.1 SP02.

### Start Script Switch for WebLogic Server 6.1 SP02 Clients

By default, a WebLogic Server 6.1 SP02 client does not interoperate with a WebLogic Server 6.1 GA or 6.1 Service Pack 1 server. This is true both for a 6.1 SP2 client and a 6.1 SP2 server acting as a client to a 6.1 GA or 6.1 SP1 server.

To allow an SP2 client to talk to a 6.1 GA or 6.1 SP1 server, add the following switch to the WebLogic Server 6.1 SP02 start script:

```
-Dweblogic.61compat=true
```

### SAX Parser and Encoding Error Detection

WebLogic Server’s built-in SAX parser now correctly parses deployment descriptor files that are written using any character set.

Prior to this fix, WebLogic Server would have failed to detect encoding problems in your `.xml` files for English-only applications. Consequently, English-only applications that contained encoding problems and ran without fail in the past may now fail with this SP02 fix; WebLogic Server will report an encoding-not-supported error.

If WebLogic Server reports this error, check the specified `.xml` file for the correct encoding name and syntax.

## Certifications

- WebLogic Server 6.1 SP02 is certified with Sun's J2SE 1.3.1\_01.
- WebLogic Server 6.1 SP02 is certified with HP JVM 1.3.1\_01. For information on the patches required by HP JVM 1.3.1\_01, see the following website:  
<http://www.hp.com/products1/unix/java/infolibrary/patches.html>
- With the release of WebLogic Server 6.1 SP02, BEA supports WebLogic jDriver for Oracle for use with Oracle 9i. You can now use the JDBC driver to connect to an Oracle 9i database. For more information about WebLogic jDriver for Oracle, see [Installing and Using WebLogic jDriver for Oracle at](#)  
<http://e-docs.bea.com/wls/docs61/oracle/index.html>.

For current information regarding platform support, see the [Supported Configurations](#) at <http://e-docs.bea.com/wls/certifications/certifications/index.html>.

## Changes to Proxy Servlets

WebLogic Server 6.1 SP02 includes new versions of the `HttpClusterServlet` and the `HttpProxyServlet` that support the HTTP 1.1 protocol. In addition, these servlets now use the same configuration parameters as the Apache, Netscape, and Microsoft IIS plug-ins.

For more information see the following documents:

- [Proxying Requests to Another HTTP Server at](#)  
[http://e-docs.bea.com/wls/docs61/adminguide/http\\_proxy.html](http://e-docs.bea.com/wls/docs61/adminguide/http_proxy.html).

- [Proxying Requests to a WebLogic Cluster at `http://e-docs.bea.com/wls/docs61/adminguide/http\_proxy\_cluster.html`.](http://e-docs.bea.com/wls/docs61/adminguide/http_proxy_cluster.html)
- [Parameters for Web Server Plug-ins at `http://e-docs.bea.com/wls/docs61/adminguide/plugin\_params.html`.](http://e-docs.bea.com/wls/docs61/adminguide/plugin_params.html)

## New Security Classes

WebLogic Server 6.1 SP02 includes two new classes related to security:

- `weblogic.security.SSL.TrustManager` allows you to override validation errors in a peer's digital certificate and continue the SSL handshake. You can also use the class to discontinue an SSL handshake by performing additional validation on a server's digital certificate chain. For more information, see the section called “Using a Trust Manager” in “[Programming with the WebLogic Security SPI](http://e-docs.bea.com/wls/docs61/security/prog.html)” at <http://e-docs.bea.com/wls/docs61/security/prog.html>.
- `SSLContext` is used to implement a secure socket protocol that holds information such as Host Name Verifier and Trust Manager for a given set of SSL connections. For more information, see the section called “Using an SSL Context” in “[Programming with the WebLogic Security SPI](http://e-docs.bea.com/wls/docs61/security/prog.html)” at <http://e-docs.bea.com/wls/docs61/security/prog.html>.

In addition, SP02 adds parameters for SSL session caching that eliminate the need for the connection to go through the SSL handshake again. For more information, see the section called “Modifying Parameters for SSL Caching” in “[Managing Security](http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html)” at <http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html>.

## Servlet and JSP Encoding Changes

As of WebLogic Server 6.1 SP02, the default encoding used for servlets and JSPs has been changed to ISO8859\_1. Previously the default encoding used was the JVM default. This change is in accordance with the latest servlet specification. Servlet Specification 2.3 SRV 4.9 states that the default encoding should be ISO8859\_1.

BEA recommends you explicitly set the encoding of the request and response objects in order to avoid any encoding problems. For further details on how to set encoding, see <http://edocs.beasys.co.jp/e-docs/wls61/jconfig/wls61jconfig.html>.

## WebLogic 6.1 SP02 and 5.1 Interoperability

As of WebLogic Server 6.1 SP02, it is possible to bi-directionally interoperate with WebLogic 5.1 Servers. This allows a WebLogic 6.1 Server to remotely invoke EJBs or RMI objects hosted on a WebLogic 5.1 Server. It also allows the reverse—a WebLogic 5.1 Server can remotely invoke EJBs or RMI objects hosted on a WebLogic 6.1 Server. For further details, see the [Interoperability Guide at <http://e-docs.bea.com/wls/docs61/interop/interop.html>](#).

## WebLogic Server 6.1 Service Pack 1

For information on problems solved in WebLogic Server 6.1 SP01, see “WebLogic Server 6.1 Service Pack 1 Solutions” on page 6-327.

## WebLogic Server 6.1 (G.A. Release)

These sections describe WebLogic Server 6.1 features, provides certification and standards support information, and highlights key known issues in the General Availability (G.A.) release.

- “Notes for WebLogic APIs” on page 1-14
- “Deprecated Features and APIs” on page 1-28
- “Documentation and Examples” on page 1-29
- “J2EE and Standards Topics” on page 1-30

- “Certifications” at <http://e-docs.bea.com/wls/certifications/certifications/index.html>

## Notes for WebLogic APIs

- “Applets” on page 1-15
- “Beanshell Code” on page 1-15
- “Clustering” on page 1-15
- “Deployment in WebLogic Server 6.1” on page 1-15
- “Enterprise Java Beans (EJBs)” on page 1-16
- “Hypertext Transfer Protocol (HTTP) 1.1” on page 1-17
- “J2EE Connector” on page 1-18
- “Java Database Connectivity (JDBC)” on page 1-19
- “JavaMail” on page 1-19
- “Java Transaction API (JTA)” on page 1-19
- “JMS” on page 1-19
- “JSP” on page 1-21
- “Security” on page 1-22
- “Servlets and Web Applications” on page 1-22
- “SNMP” on page 1-24
- “System Administration” on page 1-24
- “WebLogic Tuxedo Connector” on page 1-26
- “XML” on page 1-27
- “Web Services” on page 1-28

## Applets

A problem with the `tools.jar` shipped with the JDK prevents Applets from being served to the client correctly. If `tools.jar` is located in your server CLASSPATH, you must ensure that `weblogic.jar` appears in the server CLASSPATH before `tools.jar`.

## Beanshell Code

WebLogic Server uses an open source product called Beanshell. The initial developer of the original Beanshell code is Pat Niemeyer. Portions created by Pat Niemeyer are Copyright (C) 2000. All Rights Reserved. Beanshell is available under the Sun Public License and the GNU Lesser General Public License. The source code to Beanshell can be found at <http://www.beanshell.org>.

## Clustering

Clusters of WebLogic Servers do not require a shared network drive. Both the WebLogic Server installations and your applications can now reside on local file systems.

### Clusters, DNS, and Multihoming on NT

Be aware of naming issues when using multihoming features on Windows NT in a cluster. For example, there may be naming conflicts if a cluster is running on a multihomed Windows NT machine and one of the servers in the cluster is bound to the same DNS name as the machine name. Attempts to contact that server using the DNS name in a URL may result in Windows NT converting that DNS name to any of the IP addresses of the multihomed Windows NT machine. In this case, the request may go to the wrong address. Avoid having DNS names that match your machine name.

## Deployment in WebLogic Server 6.1

In WebLogic Server 6.1, updating an application on any single server instance to which it is targeted causes it to be updated on all servers to which is targeted. For instance, if an application is targeted to a cluster, and you update it on one of the clustered servers instances, the application will be updated on all members of cluster.

Similarly, if the application is targeted to a cluster and to a standalone server instance, updating it on the standalone server instance will result in its update on the cluster, and vice versa.

If you want to be able to update an application or component selectively on a subset of the server instances to which it is targeted, deploy unique instances of the application to different targets.

## New Deployment Tutorial Available

WebLogic Server 6.1 includes a new deployment tutorial, called “Deploying an Exploded J2EE Application”. It is available on the [WebLogic Server Samples and Tutorials](#) page.

## Enterprise Java Beans (EJBs)

WebLogic Server 6.1 includes many enhancements to WebLogic EJBs. For more information, see [Programming WebLogic Enterprise JavaBeans](#).

## EJB 1.1 and 2.0 Compliance

WebLogic Server 6.1 is compliant with the [JavaSoft EJB 1.1 Specification at http://java.sun.com/products/ejb/index.html](http://java.sun.com/products/ejb/index.html). WebLogic also contains an optional implementation of the preliminary EJB 2.0 specification. The WebLogic Server EJB documentation describes key features of the EJB 2.0 Specification that you need to understand in order to use WebLogic Server.

New EJB features in this release include:

- Non-transactional entity bean caching
- Automated Generated Primary Key Support
- Automatic table creation
- Oracle SELECT HINTS
- EJB Deployment Descriptor Editor
- EJB Client.jar support
- BLOB and CLOB support

- Cascade deletion support
- Local Interface support
- Flushing the CMP cache support
- Tuned CMP 1.1 support

## EJB Pass By Value J2EE incompatibility

Many of the WebLogic Server EJB deployment properties have default values that are optimized for performance. In some cases, these default values are not compliant with the EJB specification. Should you wish to make WebLogic Server compliant with the EJB specification, you must set the following properties:

In `weblogic-ejb-jar.xml` set “enable call by reference” to False.

In `weblogic-cmp-jar.xml`, set “include-updates” to True

In `weblogic-cmp-jar.xml`, set “check exists on method” to True.

## Hypertext Transfer Protocol (HTTP) 1.1

The following sections refer to HTTP features supported with WebLogic Server 6.1.

### Web Server

WebLogic Server 6.1 is a functional Web server that can handle high-volume Web sites, serving static HTML (text) files as well as servlets and JavaServer Pages (JSP). WebLogic Server 6.1 can also fully integrate with hardware- and software-based Web load-balancing solutions. WebLogic Server 6.1 supports the HTTP 1.1 standard.

Each WebLogic Server 6.1 hosts a default *Web server* and any number of additional Web servers that you define. Each of these additional Web servers is configured to respond to a different DNS name, in a process called Virtual Hosting.

You configure the attributes for Web servers using the new WebLogic Server Administration Console.

For more information, see [Programming WebLogic HTTP Servlets](#).

## HTTP Plug-ins for Apache, Netscape, and IIS

WebLogic Server 6.1 automatically implements keep-alive connections between the plug-ins and WebLogic Server. WebLogic Server 6.1 also supports SSL when using the HTTP plug-ins.

The Apache plug-in is now available for Apache 2.0 (non-SSL only). The Apache 2.0 version also supports HTTP 1.1 keep-alive connections.

For more information on SSL and each of the plug-ins, see the [WebLogic Server Administration Guide](#).

## HTTP Session Persistence

A new option is available for persisting HTTP sessions using cookies. For more information, see [“Configuring Session Persistence”](#).

## HTML Pages and Netscape

If you already have a Netscape browser running when you start WebLogic Server, some HTML pages will not display. For example, if you select the *About WebLogic Server* page from the Start menu, and you have Netscape running and do not have Microsoft Internet Explorer, the screen may flash but the page will not display.

## J2EE Connector

BEA WebLogic Server continues to build upon the implementation of the Sun Microsystems J2EE Platform Specification, Version 1.3 by supporting the J2EE Connector architecture. The BEA WebLogic J2EE Connector architecture enables you to connect existing legacy applications to the J2EE platform. The goal is to leverage the strengths of the J2EE platform—including component models, transaction, and security infrastructures—to address the challenges of integrating existing legacy applications.

The connector architecture defines a common interface between application servers and legacy applications, implemented in application specific resource adapters that plug in to application servers. The result is simplified enterprise application integration, using a scalable, standard architecture that leverages the benefits of the J2EE platform.

For more information, see [Programming the WebLogic J2EE Connector Architecture](#)

## Java Database Connectivity (JDBC)

The following items are new or improved JDBC features.

### MultiPools

JDBC MultiPools create a list of connection pools to be used by a single instance of WebLogic Server. A configurable algorithm determines which connection is returned. MultiPools provide support for load balancing and high availability. MultiPools make it easier for an application to switch to another RDBMS for distributed processing or during a failover situation.

For more information, see [Programming WebLogic JDBC](#).

### JDBC-based Persistence

A new column, `wl_max_inactive_interval`, has been added to the `wl_servlet_sessions` table used for JDBC-based persistence. For information on the `wl_servlet_sessions` table, see the section [“Using a Database for Persistent Storage”](#).

## JavaMail

WebLogic Server 6.1 includes an implementation of the JavaMail Specification. This is the standard reference implementation of the JavaMail Specification. For more information, see [Developing WebLogic Server Applications](#).

## Java Transaction API (JTA)

WebLogic Server 6.1 supports distributed transactions and the two-phase commit protocol with a specification-compliant JTA implementation. This implementation works with any certified XA-compliant resource such as WebLogic JMS and the WebLogic `jdbcDriver` for Oracle.

For more information, see [Programming WebLogic JTA](#).

## JMS

The following are new or improved features of JMS.

## Asynchronous JMS

Now you can disable synchronous writes to JMS File Stores. This capability results in much improved performance at the expense of reliability. If JMS reliability is not critical to your application, you may benefit from this feature. For more information, see [Programming WebLogic JMS](#).

## Durable Subscription Management

A RuntimeMBean has been added for managing JMS durable subscriptions (JMSDurableSubscriberRuntimeMBean). Use this MBean to perform simple management tasks, such as monitoring, deleting, or modifying durable subscriptions, from the WebLogic Server Administration Console.

## Redelivery Delay

You can delay the redelivery of messages when a temporary, external condition prevents an application from handling a message properly. When a message is rolled back or recovered, the RedeliveryDelay is the amount of time a message is delayed before redelivery is attempted.

If JMS immediately redelivers the message, the condition may not be clear and the application still may be unable to handle the message. Configuring an application for a RedeliveryDelay addresses this issue.

## Time To Deliver

You can schedule message deliveries to an application for specific times in the future. Message deliveries can be deferred for short periods (such as seconds or minutes) or for long stretches (such as hours later for batch processing). You can also specify a relative TimeToDeliver (in milliseconds), which WebLogic JMS will then compute into an absolute DeliveryTime for a message. Until that DeliveryTime, the message is essentially invisible until it is delivered, allowing you to schedule work at a particular time in the future.

## Redelivery Limit

You can set a limit on the number of times that WebLogic JMS will attempt to redeliver a message to an application. Once WebLogic JMS fails to redeliver a message for a specific number of times, the message can be redirected to an `ErrorDestination` queue. If no `ErrorDestination` is configured, then the message is dropped.

## JSP

The following JSP features are included with WebLogic Server 6.1:

- Options to the WebLogic JSP compiler (`weblogic.jspc`) that allow you to compile JSPs that use EJBs in Web Applications. For details, see the section [“JSP Descriptor Element”](#).
- New form validation tags are now available. The validation tags are a convenient means of validating the entries submitted from an HTML form. For more information, see [“Using JSP Form Validation Tags”](#).
- A tool to create a JSP tag library from an EJB jar file. For more information, see [“Using the WebLogic EJB to JSP Integration Tool”](#).
- JSP 1.2 Support

Support for the following features of the JSP 1.2 specification from Sun Microsystems has been added. *Version 1.2 of the specification is a proposed final draft of the specification and is subject to change. If you are planning to use JSP 1.2 features in your application, note that the specification has not been finalized and could change in the future.*

- `<jsp:include flush="false">`. Allows you to control when the buffer is flushed for an included JSP. For more information, see the section [“Including Requests”](#).
- `TryCatchFinally` interface for a JSP tag. For more information, see [Handling Exceptions within a Tag Body](#).
- Tag Library Validator allows you to write a class to validate a JSP. For more information, see [Using a Tag Library Validator](#).
- `IterationTag` class. For more information, see [Iteration Over a Body Tag](#).
- Definition of variables in the tag library descriptor. For more information see [Defining Scripting Variables](#).

## Security

WebLogic Server 6.1 security functionality includes:

- JAAS login and realm-based authentication support
- Enhanced support for generating audit trails
- ACL improvements
- Password-guessing protection
- Denial-of-service protection
- Improved administration of the RDBMS realm

For more information, see [Programming WebLogic Security](#).

## Default and Custom Login Modules

WebLogic Server uses the default `LoginModule` (`weblogic.security.internal.ServerLoginModule`) to gather authentication information during server initialization. To replace the default Login module, edit the `server.policy` file and replace the name of the default Login module with the name of a custom Login module.

Optionally, custom login modules can be specified in the `server.policy` file ahead of the default `LoginModule`. The JAAS implementation in WebLogic Server uses Login modules in the order in which they are defined in the `server.policy` file. The default Login module checks for existing system user authentication definitions prior to execution and does nothing if they are already defined.

The default Login module is required to define JVM properties for both the system username and password. These properties are specified as `weblogic.management.username` and `weblogic.management.password` respectively. In order to use a custom Login module, these properties must be set accordingly.

## Servlets and Web Applications

The following new features and changes apply to servlets and Web Applications:

- Web Application Events (Servlet 2.3 only)

Web Application Events provide notifications of a change in state of the *Servlet context* (each Web Application uses its own servlet context) or of an *HTTP session object*. You can write event listener classes that respond to these changes in state. For more information, see [Using Application Events and Listeners](#).

■ Filters (Servlet 2.3 only)

A filter is a Java class that is invoked in response to a request for a resource in a Web Application. These resources can include Java Servlets, JavaServer Pages (JSPs), or static resources such as HTML pages or images. A filter intercepts the request and can examine and modify the response and request objects or execute other tasks. For more information, see [Using Filters](#).

■ Single sign-on between Web Applications. This means the server tracks users at the Web Server level and not the Web Application level, as mandated by the Servlet 2.2 specification.

■ The format for cookies:

- Cookies created by a WebLogic Server 6.0 without a Service Pack will not be recognized by WebLogic Server 6.1.
- Cookies no longer include the context path of a Web Application. This change allows for a single authentication to be used for all Web Applications in a domain. If you need a Web Application to use a unique authentication or to generate a cookie that is unique to the Web Application, you can specify a cookie name for each Web Application in the `weblogic.xml` deployment descriptor. For more information, see [Multiple Web Applications and Cookies](#).
- The delimiter used in the cookie has changed from `"/"` and `"|"` to `"!"`. The `"/"` delimiter caused URL rewriting to fail and the `"|"` delimiter caused Wireless Application Protocol (WAP) gateways to fail. Both URL rewriting and WAP now function correctly with WebLogic Server.

Version 2.3 of the Servlet Specification from Sun Microsystems is a proposed final draft of the servlet specification and is subject to change. If you are planning to use any features introduced with version 2.3 in your application, note that the specification has not been finalized and could change in the future.

## SNMP

WebLogic Server 6.1 can function as a Simple Network Management Protocol (SNMP) agent. The WebLogic SNMP agent runs as a service that responds to requests from SNMP managers and sends SNMP trap notifications to SNMP managers. The WebLogic SNMP agent uses standard Java Management Extension (JMX) interfaces to access WebLogic resources. For more information, see the [WebLogic SNMP Management Guide](#).

## System Administration

### Installation

WebLogic Server 6.1 includes an installation program that makes it easier to install WebLogic Server on both Windows and UNIX systems. The installer unpacks the distribution, performs basic configurations, and sets up shortcuts for using WebLogic Server. In addition, the JDK is included in the package so that the server is ready to run.

For details, see the [WebLogic Server Installation Guide](#).

### Internationalization

WebLogic Server 6.1 can deliver content in any language, including those languages that require double-byte character sets. For information about using the new internationalization API, see the [WebLogic Server Internationalization Guide](#).

In addition, a Kanji version of WebLogic Server 6.1 will be available. For more information, contact your BEA sales representative.

### Administration Console

Improvements to management architecture enable you to make dynamic configuration changes to running instances of WebLogic Server. A Web-based Administration Console is your window into the WebLogic Administration Service, an implementation of the Java Management Extension (JMX) standard. Using the Administration Console, you can configure attributes, deploy applications and components, monitor resource usage, view log messages, edit deployment descriptors and `.xml` files, and perform other management activities. Administration Console features include:

- Browser interface

- Centralized user and security management
- Dynamic configuration management
- Editing tool for deployment descriptors of J2EE Archives
- Administrative domains
- Management of one or more clusters
- A centralized application repository
- Metrics on the status of servers
- Centralized access to log messages for all servers in a domain

For more information, see the [WebLogic Server Administration Guide](#).

## Production Mode and Development Mode

There is now a flag that can be used to switch between production and development modes. When the server starts, it loads and deploys any applications that have configuration information in the `config.xml` file, whether you are using production or development mode. If you are running in development mode, the server will also deploy or redeploy any applications stored or placed in the `.\applications` directories after startup. This is called dynamic deployment and is useful when you are developing your applications. If you shut down the server, the configuration information for any applications that were dynamically deployed is written to the `config.xml` file in the application's domain. The production/development flag can be added to the startup script for your domain. If it is not added, development mode will be activated. To change the mode:

1. With a text editor, open the start script for your domain.
2. Edit the line that begins `set STARTMODE=` to add the value `true` or `false`. `True` is for production mode, and `false` is for development mode.

## Tools

WebLogic Server 6.1 includes the `weblogic.refresh` tool, which lets you refresh static components of an application without redeploying the application. Use `weblogic.refresh` to refresh, add, or delete static files such as:

- JSPs

- XML files
- HTML files
- Image files such as .gif and .jpg
- Text files

## **WebLogic Tuxedo Connector**

In WebLogic Server Release 6.1, the WebLogic Tuxedo Connector has the following limitations:

- Does not support dynamic configuration changes to WebLogic Tuxedo Connector gateway
- Does not support View buffers.
- Does not support in-bound RMI/IIOP transactions from a Tuxedo application.
- Does not support clustering. Only one instance of WebLogic Tuxedo Connector can be used in a clustered environment.
- Does not support Tuxedo 6.5 running on VMS, AS/400, and OS/390 platforms.

## **WebLogic-Tuxedo Interoperability**

WebLogic Tuxedo Connector (WTC) provides interoperability between WebLogic Server applications and Tuxedo ATMI, CORBA Java, and CORBA C++ server applications. WTC tBridge functionality provides Tuxedo /Q and JMS advanced messaging services. For more information, see the [WebLogic Tuxedo Connector Guide](#).

## **BEA WebLogic C++ Client**

WebLogic Server 6.1 SP3 interoperates with C++ clients using the Tuxedo 8.0 C++ Client ORB. Tuxedo release 8.0 RP 56 and above is required. WebLogic Server users should contact their BEA Service Representative for information on how to obtain the Tuxedo C++ Client ORB. For more information on how WebLogic Server interoperates with a Tuxedo C++ Client ORB, see [RMI-IIOP with Tuxedo and Tuxedo Clients](#).

## XML

WebLogic Server 6.1 supports XML as an essential component. JSPs can be used to generate and consume XML between servers or between a server and clients. WebLogic Server 6.1 supports XSL processing tags for JSPs. EJBs use XML to describe deployment properties, which provides data portability. The server provides an XML schema repository for DTDs, managed by the Administration Console.

The XML subsystem of WebLogic Server 6.1 includes the following new features and improvements:

- The JAXP API is upgraded to 1.1.
- WebLogic server includes a high-performance XML parser, specifically designed to parse small-to-medium sized XML documents. This XML parser is non-validating and can be used only in SAX mode.
- The built-in Apache Xerces parser is upgraded to 1.3.1.
- The built-in Apache Xalan transformer is upgraded to 2.0.1.
- The XML Registry is divided into two distinct sections: one to configure parsers and transformers and one to configure external entity resolution. As a result, the XML Registry is easier and more intuitive to use.
- The XML Registry now includes the ability to cache external entities accessed via URLs on the local computer, in addition to copying entities locally. The external entity cache can be configured and monitored using the XML Registry.
- The 6.0 XML feature of generating custom parsers is deprecated. Instead, programmers should use the WebLogic high-performance XML parser where previously they used custom generated parsers.

For more information on XML, see [Programming WebLogic XML](#).

## XML Editor

WebLogic Server now provides a simple, user-friendly tool for creating and editing XML files. It can validate XML code according to a specified DTD or XML Schema. The XML editor can be used on Windows or Solaris machines and can be downloaded from [dev2dev Online](#).

## Web Services

The current WebLogic Server™ release includes a new feature: WebLogic Web Services. Web services are an emerging family of Web/XML standards and conventions that enable loosely coupled application-to-application interoperability across programming models. The Web services implementation in WebLogic Server 6.1 provides simple extensions to the J2EE programming model that enable J2EE applications to be exposed as Web services. WebLogic Web Services are standards-based and interoperate with other Web services hosted on non-WebLogic servers. Both Java and non-Java client applications (such as Microsoft SOAP Toolkit clients) can invoke WebLogic Web Services. For detailed information about developing and invoking WebLogic Web services, see Programming WebLogic Server WebServices.

## Deprecated Features and APIs

The following features are deprecated in WebLogic Server 6.1 and will not be supported in the future:

- `weblogic.db.jdbc.EventfulTableDataSet`
- WebLogic WorkSpaces
- WebLogic Events
- WebLogic Time
- WebLogic JDBC t3 driver
- ZAC

For a list of deprecated WebLogic Server classes, see <http://e-docs.bea.com/wls/docs61/javadocs/deprecated-list.html>.

## Documentation and Examples

The examples included with WebLogic Server 6.1 now include `ant` build scripts. These scripts accompany the examples as `build.xml` files and can be run on any platform supported by WebLogic Server 6.1. Execute these build scripts on a command line by locating the proper directory for that example and typing:

```
ant
```

The following documentation and examples are new to WebLogic Server 6.1:

- New documentation for the `config.xml` file is in [WebLogic Server Configuration Reference](#).
- Performance and tuning information for WebLogic Server 6.1 is in the [BEA WebLogic Server Performance and Tuning Guide](#).
- WebLogic Web Services documentation and examples are in the [Programming WebLogic Web Services Guide](#).
- WebLogic Server 6.1 can function as a Simple Network Management Protocol (SNMP) agent. See the new [SNMP Management Guide](#).
- JCA examples and documentation on [Managing the WebLogic J2EE Connector Architecture](#) are provided in the [WebLogic Server Administration Guide](#).
- [Assembling and Configuring Web Applications](#) provides all the configuration, assembly, and deployment information for Web Applications in one book. Some of this information was previously located in the [Administration Guide](#) and in [Developing WebLogic Server Applications](#).
- Several examples demonstrating wireless connectivity are available with WebLogic Server 6.1.
- An extensive migration tutorial demonstrates how applications deployed on earlier releases of WebLogic Server can be adapted to WebLogic Server 6.1.
- A new example package demonstrates Oracle's delete cascade. It uses the one-to-many relationships between two EJBs written to the EJB 2.0 specification.
- Auto Primary Key Generation EJB examples have been added. These examples automatically generate primary keys during insertions into a database table.

- For documentation and examples, see the [WebLogic Tuxedo Connector](#) documentation.

## **WebLogic Server Tour**

The WebLogic Server Tour is revised. It provides an overview of WebLogic Server using the Pet Store application to demonstrate features. The Tour is available from the Start menu. The Pet Store is now based on version 1.1.2 of Sun's Java Pet Store and has been upgraded to use build scripts with Ant 1.3.

## **J2EE and Standards Topics**

### **Java Development Kit**

A JDK provides a Java runtime environment (the Java Virtual Machine or JVM) and tools for compiling and debugging your Java applications. Your installation of WebLogic Server 6.1 includes the 1.3.1rc2 JDK from Sun Microsystems.

BEA WebLogic Server 6.1 is the first e-commerce transaction platform to implement advanced J2EE 1.3 features. To comply with the rules governing J2EE, BEA Systems provides two separate downloads: one with J2EE 1.3 features enabled, and one that is limited to J2EE 1.2 features only. Both downloads offer the same container and differ only in the APIs that are available.

### **WebLogic Server 6.1 with J2EE 1.2 Plus Additional J2EE 1.3 Features**

With this download, WebLogic Server defaults to running with J2EE 1.3 features enabled. These features include EJB 2.0, JSP 1.2, Servlet 2.3, and J2EE Connector Architecture 1.0. When you run WebLogic Server 6.1 with J2EE 1.3 features enabled, J2EE 1.2 applications are still fully supported. The J2EE 1.3 feature implementations use non-final versions of the appropriate API specifications. Therefore, application code developed for BEA WebLogic Server 6.1 that uses the new features of J2EE 1.3 may be incompatible with the J2EE 1.3 platform supported in future releases of BEA WebLogic Server.

The sections below list the J2EE 1.3 API classes and deployment descriptors that are not implemented in WebLogic Server 6.1.

## Enterprise Java Beans Deployment Descriptors

WebLogic Server 6.1 fully implements all EJB 2.0 deployment descriptors defined in Sun Microsystem's `ejb-jar.xml` document type definitions (DTD) file with the following exceptions:

- `exclude_list`
- `unchecked`
- `run_as`

**Note:** The `ejb-jar.xml` document type definitions (DTD) file is part of Sun Microsystem EJB 2.0 specification.

## JDBC API Classes

API	Class or Interface	Methods not Implemented	Notes
<code>java.sql</code>	<code>CallableStatement</code>	<code>getResultSet(int parameterIndex)</code>	Implemented as a method for the OracleCallableStatement Oracle extension; you must use a JDBC driver that supports the OracleCallableStatement extension, such as the Oracle Thin Driver.
	<code>Driver</code>	<code>getPropertyInfo(java.lang.String url, java.util.Properties info)</code>	Not Implemented

## JSP 1.2 API Classes

API	Class	Differs from J2EE 1.3
<code>javax.servlet.jsp</code>	<code>java.beans.Beans</code>	The <code>jsp:id</code> mechanism has not been implemented

**Note:** The following feature has not been implemented:

A JAR containing a packaged tag libraries can be dropped into the `WEB-INF/lib` directory to make its classes available at request time.

## JSP 1.2 Deployment Descriptors

- The `<listener>` element in the `taglib.tld` is not registered with the web application.
- The `<example>` element in the `taglib.tld` is not honored.
- WebLogic Server 6.1 uses the older signature of the `TaglibraryValidator.validate()` method which returns a string

---

## Servlet API Classes

API	Class	Differs from J2EE 1.3
javax.servlet	Filter	The Filter interface that we support has following methods: doFilter(ServletRequest, ServletResponse, FilterChain) getFilterConfig() Return the FilterConfig for this Filter. setFilterConfig(FilterConfig) The container calls this method when the Filter is instantiated and passes in a FilterConfig
	ServletContext	Added the directory argument (String) in the ServletContext.getResourcePaths() method
	ServletContextAttributesListener	The name in the final specification has been changed to: ServletContextAttributeListener
javax.servlet.http	HttpServletResponse	Added the status code 307 (temporary redirect) to HttpServletResponse
	HttpSession	The getServletContext() method has been added to HttpSession
	HttpSessionAttributesListener	The name in the final specification has been changed to: HttpSessionAttributeListener

## Servlet Deployment Descriptors

WebLogic Server 6.1 fully implements all Servlet 2.3 deployment descriptors with the following exceptions:

- `ejb-local-ref`
- `run_as`

## WebLogic Server 6.1 with J2EE 1.2 Certification

With this download, WebLogic Server defaults to running with J2EE 1.3 features disabled and is fully compliant with the J2EE 1.2 specification and regulations.

## Product CD Installers for J2EE 1.2 and 1.3

In addition to being available at <http://commerce.bea.com/downloads/products.jsp>, both distributions are provided on the WebLogic Server 6.1 product CD. (On Windows machines, the installer for WebLogic Server with J2EE 1.3 features enabled starts automatically when you insert the CD.)

## Standards Support

Standard	Version
HTTP	1.1
J2EE Connector Architecture	1.0
J2EE EJB	2.0
J2EE JDBC	2.0
J2EE JNDI	1.2
J2EE JSP	1.1
J2EE JTA	1.0.1
J2EE JMS	1.0.2
J2EE RMI	1.0
RMI/IIOP	1.0
J2EE Servlet	2.2
LDAP	2
SSL	3
X.509	3

# 2 Upgrading WebLogic Server 5.1 to Version 6.1

The following sections contain information necessary to upgrade your system from WebLogic Server 5.1 to WebLogic Server 6.1:

- Upgrading Your WebLogic Server Configuration: Main Steps
- Converting the weblogic.properties File
- Classloading in WebLogic Server 6.x
- Modifying Startup Scripts
- Upgrading Oracle
- Upgrading JMS
- Upgrading JVM

# Upgrading Your WebLogic Server Configuration: Main Steps

Take the following steps to upgrade from WebLogic Server 5.1 to WebLogic Server 6.1:

1. Install WebLogic Server 6.1. See the *WebLogic Server Installation Guide* at <http://e-docs.bea.com/wls/docs61/install/index.html>.
2. Convert your `weblogic.properties` file. For instructions on how to convert your `weblogic.properties` file, see “Converting the `weblogic.properties` File” on page 2-2 and the *Console Help* documentation at <http://e-docs.bea.com/wls/docs61/ConsoleHelp/conversion.html>.
3. Add classes to your Java system CLASSPATH. For more information see “Classloading in WebLogic Server 6.x” on page 2-3.
4. Modify your startup scripts to work with WebLogic 6.1. See “Modifying Startup Scripts” on page 2-4.
5. Upgrade JMS. Many new configuration attributes have been added to JMS since WebLogic Server 5.1. For more information, see “Upgrading JMS” on page 2-5.
6. Upgrade your JVM. WebLogic Server 6.1 requires JDK 1.3. For more information, see the *Platform Support Page* at [www.weblogic.com/platforms/index.html](http://www.weblogic.com/platforms/index.html).

## Converting the `weblogic.properties` File

In earlier versions of WebLogic Server, configuration properties were stored in the `weblogic.properties` file. In WebLogic Server 6.0 and later, configuration attributes for the server are stored in a persistent `.xml` file, `config.xml`. Configuration attributes for applications are stored in `.xml` files specific to that particular application. Depending on your application you may have `web.xml`,

`weblogic.xml`, or `application.xml` files associated with your application. The `weblogic.properties` file that handled configuration in earlier releases of WebLogic is not used in WebLogic Server 6.0 and later.

Convert your existing `weblogic.properties` file to the appropriate `.xml` files using the Administration Console. Procedures for converting your `weblogic.properties` file are provided in the *Console Help* documentation at <http://e-docs.bea.com/wls/docs61/ConsoleHelp/conversion.html>.

- Do not edit the `config.xml` file directly. Access to the configuration is provided through the Administration Console, a command line utility, or programmatically through the configuration API. For details on configuring WebLogic Server, refer to the *Administration Guide* at <http://e-docs.bea.com/wls/docs61/adminguide/config.html>.
- Security properties are stored in the `fileRealm.properties` file.
- The `weblogic.common.ConfigServicesDef` API, which provided methods to get properties out of the `weblogic.properties` file, has been removed from this version.

## Classloading in WebLogic Server 6.x

Earlier versions of WebLogic Server used the `WebLogic` classpath property (`weblogic.class.path`) to facilitate dynamic class loading. In WebLogic 6.0 and later, the `weblogic.class.path` is no longer needed. You can now load classes from the Java system classpath.

To include the classes that formerly were specified in `weblogic.class.path` in the standard Java system classpath, use the `-classpath` option on the command line, or set the `CLASSPATH` environment variable.

## Modifying Startup Scripts

If you used WebLogic Server startup scripts with a previous version of the product, modify them to work with 6.1.

- Modify the startup scripts as described in *Setting the Classpath Option* at <http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#settingclasspath>. The WebLogic classpath is no longer used; use the Java system classpath as described in the preceding section, “*Classloading in WebLogic Server 6.x*”.
- It is no longer necessary to include the license file in the classpath.
- With the new management system, there is a distinction between an Administration Server and Managed Servers. Consequently, scripts that start servers must be rewritten according to how you plan to administer your servers. For the new commands and their required arguments, see *Starting and Stopping WebLogic Servers* at <http://e-docs.bea.com/wls/docs61/adminguide/startstop.html>.

## Upgrading Oracle

BEA, mirroring Oracle's support policy, supports the Oracle releases called out in *Installing and Using WebLogic jDriver for Oracle* at <http://e-docs.bea.com/wls/docs61/oracle/index.html>. BEA no longer supports the following Oracle client versions: 7.3.4, 8.0.4, 8.0.5, and 8.1.5.

If you want to use the Oracle Client Version 7.3.4, you can use the backward compatible oci816\_7 shared library. As stated above, BEA no longer supports this configuration.

If you want to upgrade to Oracle Client Version 8.1.7, or read detailed documentation on the WebLogic jDriver and Oracle databases, see *Installing and Using WebLogic jDriver for Oracle* at <http://e-docs.bea.com/wls/docs61/oracle/index.html>.

For supported platforms, as well as DBMS and client libraries, see the BEA *Platform Support Page* at <http://www.weblogic.com/platforms/index.html>. The most current certification information will always be posted on the Platform Support page.

## Upgrading JMS

The following table lists the WebLogic JMS configuration attributes that have been added since WebLogic Server 5.1.

## 2 Upgrading WebLogic Server 5.1 to Version 6.1

---

For information on migrating an existing application, see *Migrating WebLogic Server Applications* at <http://e-docs.bea.com/wls/docs61/jms/migrat.html>.

Component	JMS Attribute	Description
JMS Connection Factories	Message Maximum	Maximum number of messages that can exist for an asynchronous session and that have not yet been passed to the message listener. This attribute defaults to 10.
	Overrun Policy	Overrun policy for multicast sessions. When the number of outstanding messages reaches the Messages Maximum attribute value, messages are discarded based on the specified policy.  If set to KeepNew, the most recent messages are given priority over the oldest messages, and the oldest messages are discarded, as needed. If set to KeepOld, the oldest messages are given priority over the most recent messages, and the most recent messages are discarded, as needed.  This attribute defaults to KeepOld.
	Allow Close In OnMessage	Flag for specifying whether a connection factory creates message consumers that allow a <code>close()</code> method to be issued within its <code>onMessage()</code> method call. This attribute is disabled by default.
	Transaction Timeout	Timeout value (in milliseconds) for transacted sessions. This attribute defaults to 3600.
	User Transactions Enabled	Flag for specifying whether a connection factory creates sessions that are JTA aware. This attribute is disabled by default.
	Default Redelivery Delay	Default time delay, in milliseconds, before rolled back or recovered messages are redelivered. This attribute defaults to 0.

Component	JMS Attribute	Description
	Default Time To Deliver	Default time delay, in milliseconds, between when a message is produced and when it is made visible on its destination. This attribute defaults to 0.
	XA Connection Factory Enabled	Flag for specifying whether or not an XA queue or XA topic connection factory is returned, instead of a queue or topic connection factory. It can be used to return a XA queue session or XA topic session, which has the <code>getXAResource</code> method.
	Acknowledge Policy	<p>The message acknowledge policy for the connection factory. This attribute only applies to implementations that use the <code>CLIENT_ACKNOWLEDGE</code> acknowledgement mode for a non-transacted session. Valid values are:</p> <ul style="list-style-type: none"> <li>■ All—acknowledge all messages ever received by a given session, regardless of which message calls the acknowledge method.</li> <li>■ Previous—acknowledge all messages received by a given session, but only up to and including the message that calls the acknowledge method.</li> </ul> <p>This attribute defaults to All.</p>
JMS Servers	JMS Default Connection Factories Enabled	Flag specifying whether the JMS default connection factories are instantiated on the JMS server. This attribute is enabled by default.

Component	JMS Attribute	Description
	Bytes Paging Enabled	<ul style="list-style-type: none"> <li>■ If the Bytes Paging Enabled check box is not selected (False), then server bytes paging is explicitly disabled.</li> <li>■ If the Bytes Paging Enabled check box is selected (True), a paging store has been configured, and both the Bytes Threshold Low and Bytes Threshold High attributes are greater than -1, then server bytes paging is enabled.</li> <li>■ If either the Bytes Threshold Low or Bytes Threshold High attribute is undefined, or defined as -1, then server bytes paging is implicitly disabled—even though the Bytes Paging Enabled check box is selected (True).</li> </ul>
	Messages Paging Enabled	<ul style="list-style-type: none"> <li>■ If the Messages Paging Enabled check box is not selected (False), then server messages paging is explicitly disabled.</li> <li>■ If the Messages Paging Enabled check box is not selected (False), then server messages paging is explicitly disabled.</li> <li>■ If either the Messages Threshold Low or Messages Threshold High attribute is undefined, or defined as -1, then server paging is implicitly disabled—even though the Messages Paging Enabled check box is selected (True).</li> </ul>
	Paging Store	<p>The name of the persistent store where non-persistent messages are paged. A paging store cannot be the same store used for persistent messages or durable subscribers. Two JMS servers cannot use the same paging store; therefore, you must configure a unique paging store for each server.</p>

Component	JMS Attribute	Description
JMS Destinations	Enable Store	<p>Flag specifying whether the destination uses the backing store specified by the JMS server.</p> <p>If this flag is enabled, but no backing store is defined, then the configuration fails and WebLogic JMS does not boot. If this flag is disabled, then the destination does not support persistent messages. If this flag is set to Default, then the destination uses the backing store if one is defined.</p> <p>This attribute defaults to Default.</p>
	Multicast Address	<p>IP address used for multicasting. This address is used to transmit messages to multicast consumers. This attribute has no default.</p>
	Multicast TTL	<p>Time-to-live value used for multicasting, specifying the number of routers that the message can traverse en route to the consumers. This attribute defaults to 0.</p>
	Multicast Port	<p>IP port used for multicasting. This port is used to transmit messages to multicast consumers. This attribute defaults to 6001.</p>
	Time To Deliver Override	<p>Defines the default delay, in milliseconds, between when a message is produced and when it is made visible on its destination, regardless of the delivery time specified by the producer and/or connection factory. The default value (-1) specifies that the destination will not override the Time To Deliver setting.</p>
	Delivery Mode Override	<p>Delivery mode assigned to all messages that arrive at the destination regardless of the delivery mode specified by the message producer. The default value of No-Delivery specifies that the delivery mode will not be overridden.</p>

<b>Component</b>	<b>JMS Attribute</b>	<b>Description</b>
	Redelivery Limit	<p>The number of redelivery tries a message can have before it is placed in the error destination. Depending on whether an error destination is configured, the following occurs when the redelivery limit is reached:</p> <ul style="list-style-type: none"><li>■ If no error destination is configured, or the quota for the error destination would be exceeded, then persistent and non-persistent messages are simply dropped.</li><li>■ If an error destination is configured and the error destination is at quota, then an error message is logged and the message is dropped. However, if the message is persistent, it remains in the persistent store. This ensures that a persistent message will be redelivered when WebLogic Server is rebooted.</li></ul> <p>The default value (-1) specifies that the destination will not override the Redelivery Limit setting.</p>
	Error Destination	<p>A destination for messages that have reached their redelivery limit. If the error destination is null, then such messages are simply dropped. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. This attribute defaults to none.</p>

Component	JMS Attribute	Description
	Bytes Paging Enabled	<ul style="list-style-type: none"> <li data-bbox="767 256 1188 337">■ If Bytes Paging Enabled is set to False, then destination-level bytes paging is disabled for this destination.</li> <li data-bbox="767 354 1188 548">■ If Bytes Paging Enabled is set to True, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attributes are greater than -1, then destination-level bytes paging is enabled for this destination.</li> <li data-bbox="767 565 1188 732">■ If Bytes Paging Enabled is set to Default, then this value inherits the template's value—if a template is specified. If no template is configured for the destination, then the Default value is equivalent to False.</li> </ul>
	Messages Paging Enabled	<ul style="list-style-type: none"> <li data-bbox="767 764 1188 846">■ If Messages Paging Enabled is set to False, then destination-level messages paging is disabled for this destination.</li> <li data-bbox="767 862 1188 1057">■ If Messages Paging Enabled is set to True, a paging store has been configured for the JMS Server, and both the Messages Threshold Low and Messages Threshold High attributes are greater than -1, then destination-level messages paging is enabled for this destination.</li> <li data-bbox="767 1073 1188 1240">■ If Messages Paging Enabled is set to Default, then this value inherits the template's value—if a template is specified. If no template is configured for the destination, then the Default value is equivalent to False.</li> </ul>

Component	JMS Attribute	Description
JMS Template	Bytes Paging Enabled	<ul style="list-style-type: none"> <li>■ If the Bytes Paging Enabled check box is not selected (False), then destination-level bytes paging is disabled for the JMS template's destinations—unless the destination setting overrides the template.</li> <li>■ If the Bytes Paging Enabled check box is selected (True), a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attributes are greater than -1, then destination-level bytes paging is enabled for the JMS template's destinations—unless the destination setting overrides the template.</li> <li>■ If no value is defined in the JMS Template MBean, then the value defaults to False and bytes paging is disabled for the JMS template's destinations.</li> </ul>
	Messages Paging Enabled	<ul style="list-style-type: none"> <li>■ If the Messages Paging Enabled check box is not selected (False), then destination-level messages paging is disabled for the template's destination—unless the destination setting overrides the template.</li> <li>■ If the Messages Paging Enabled check box is selected (True), a paging store has been configured for the JMS Server, and both the Messages Threshold Low and Messages Threshold High attributes are greater than -1, then destination-level messages paging is enabled for this destination—unless the destination setting overrides the template.</li> <li>■ If no value is defined in the JMS Template MBean, then the value defaults to False and messages paging is disabled for the template's destinations.</li> </ul>

# JSP

The behavior of the JSP include directive has changed between WebLogic Server 5.1 and the current version. In versions through WebLogic Server 5.1, the JSP include directive logged a Warning-level message if it included a non-existent page. In WebLogic Server 6.0 and later, it reports 500 Internal Server Error in that case. You can avert the error by placing an empty file at the referenced location.

## Upgrading JVM

To run WebLogic Server 6.1, you **must** upgrade to JDK 1.3. Only Java client applications can use JDK 1.2 and above. The latest information regarding certified JVMs is available at the *Platform Support Page* at [www.weblogic.com/platforms/index.html](http://www.weblogic.com/platforms/index.html).



# 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

- The following sections describe how to migrate applications from WebLogic Server Version 4.5 or 5.1 to WebLogic Server 6.x, and provide additional information concerning migration and deployment. Instructions apply to migrations from both WebLogic Server 4.5 and 5.1 to both WebLogic Server 6.0 and 6.1. For a more detailed procedure, see the [Migration Tutorial](#) on *BEA dev2dev*.
- The weblogic.properties File and .xml Files
- WebLogic Server 6.x Application Types
- Migration Main Steps
- Converting the weblogic.properties File to .xml files
- weblogic.properties Mapping Table
- Migrating Web Applications
- Migrating and Converting Enterprise JavaBeans Applications
- Creating an Enterprise Application
- Understanding J2EE Client Applications
- Migrating JMS Applications
- Additional Migration and Deployment Considerations

# The `weblogic.properties` File and `.xml` Files

Earlier releases of WebLogic Server used a `weblogic.properties` file to configure applications. In WebLogic Server 6.0 and 6.1, configuration of applications is handled through XML descriptor files and the Administration Console. Converting a `weblogic.properties` file from an earlier version of WebLogic Server creates a new domain for your applications and adds `.xml` files that define how your applications are set up.

The `config.xml` file is an XML document that describes the configuration of an entire WebLogic Server domain. The `config.xml` file consists of a series of XML elements. The Domain element is the top-level element, and all elements in the Domain are children of the Domain element. The Domain element includes child elements, such as the Server, Cluster, and Application elements. These child elements may have children themselves. Each element has one or more configurable attributes. An attribute defined in `config.dtd` has a corresponding attribute in the configuration API.

The `weblogic.xml` file contains WebLogic-specific attributes for a Web application. You define the following attributes in this file: HTTP session parameters, HTTP cookie parameters, JSP parameters, resource references, security role assignments, character set mappings, and container attributes.

The deployment descriptor `web.xml` is defined by the servlet 2.3 specification from Sun Microsystems. This deployment descriptor can be used to deploy a Web Application on any J2EE-compliant application server.

## WebLogic Server 6.x Application Types

Applications on J2EE-compliant servers such as WebLogic Server 6.0 and 6.1 are created and deployed as one of the following four types: Web Applications, Enterprise JavaBeans, Enterprise Archives, and Client Applications. To migrate your existing components to WebLogic Server 6.1, you create the appropriate J2EE deployment units. Web Applications are usually a collection of servlets, JSPs, and HTML files,

packaged as `.war` files. Enterprise JavaBeans (packaged as `.jar` files) are server-side Java components written according to the EJB specification. Enterprise Archives (`.ear` files) can contain a combination of EJB components and Web Application components. Client Applications are Java classes that connect to WebLogic Server using Remote Method Invocation (RMI). Each of the aforementioned J2EE deployment units are discussed in greater detail in the following sections.

## Migration Main Steps

The following steps outline the general migration procedure. Later sections describe these steps in more detail as they apply to particular types of applications. Also see the [Migration Tutorial](#) on *BEA dev2dev*.

1. Convert your `weblogic.properties` file to `.xml` files.
2. In WebLogic Server 6.x, specify your new domain. You will migrate applications to this domain.
3. Modify your start and setenv scripts to point to your new domain.
4. Restart WebLogic Server 6.x.
5. Migrate your application.

## Converting the `weblogic.properties` File to `.xml` files

**Note:** In order for the conversion to work properly, you need to specify the following in your `weblogic.properties` file:

```
weblogic.password.system=gumby1234
```

Convert your `weblogic.properties` file to the appropriate `.xml` files by following these steps:

1. Start the default WebLogic Server 6.1 server and the default WebLogic Server 6.1 Administration Console. For information on starting WebLogic Server 6.1, see [Post-Installation Tasks at `http://e-docs.bea.com/wls/docs61/install/instpos.html`](http://e-docs.bea.com/wls/docs61/install/instpos.html).
2. At the home page for the WebLogic Administration Console (for example: `http://localhost:7001/console/index.jsp`) click on the “Convert `weblogic.properties`” link under the heading Getting Started.
3. Use the Console’s links to navigate the server's file system and find the root directory of your previous version of WebLogic Server (for example: `C:/weblogic`). When you have found the correct directory, click on the icon next to it to select it.
4. If you have additional per server `weblogic.properties` files or clustering `weblogic.properties` files, select them using the provided windows. **If you have chosen the correct root directory of your previous version of WebLogic Server, your global `weblogic.properties` file will be converted regardless of any additional properties files that you select.**
5. Enter a name for your new domain in the provided window in the Console. Click Convert.

The process of converting your `weblogic.properties` file creates the `config.xml` file in the `wlserver6.1/config/domainName` directory. This file contains configuration information specific to your domain. It creates the web application in the `wlserver6.1/config/domainName/production_apps/DefaultWebApp_serverName/` directory. The `web.xml` and `weblogic.xml` files for the web application are placed inside the `wlserver6.1/config/domainName/production_apps/DefaultWebApp_serverName/WEB-INF/` directory.

**Note:** The conversion utility described above specifies the Java home location in the `weblogic.xml` file. It reads this location using the `System.getProperty(java.home)`, which means that it will specify the Java home location on which WebLogic Server was started for the conversion.

Throughout this document, the directory of the new domain that has been created is referred to as `domainName`. The default domain that is provided with the original installation of WebLogic Server 6.1 is called `mydomain` and is located in the `wlserver6.1/config/` directory.

The startup scripts, which are generated when a `weblogic.properties` file is converted, are named `startdomainName.cmd` (for Windows users) and `startdomainName.sh` (for UNIX users) and exist under the `WL6x_HOME/config/domainName` directory in your WebLogic Server 6.x distribution. These scripts will start the server in the new domain.

See [Starting and Stopping the WebLogic Server at `http://e-docs.bea.com/wls/docs61/adminguide/startstop.html`](#) for more information on scripts and starting servers.

## weblogic.properties Mapping Table

The `weblogic.properties` mapping table shows which `config.xml`, `web.xml`, or `weblogic.xml` attribute handles the function formerly performed by `weblogic.properties` properties. If the attribute can be configured in the Administration Console, the table shows the navigation path to the attribute.

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
<code>weblogic.administrator.email</code>	<code>config.xml:EmailAddress</code> (Administrator element)	
<code>weblogic.administrator.location</code>	<code>config.xml:Notes</code> (freeform, optional) (Administrator element)	
<code>weblogic.administrator.name</code>	<code>config.xml:Name</code> (Administrator element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.administrator.phone	config.xml: PhoneNumber (Administrator element)	
weblogic.cluster.defaultLoadAlgorithm	config.xml: DefaultLoadAlgorithm	Clusters: <i>clustername</i> : Configuration: General: Default Load Algorithm
weblogic.cluster.multicastAddresses	config.xml: MulticastAddress	Clusters: <i>clustername</i> : Configuration: Multicast: Multicast Address
weblogic.cluster.multicastTTL	config.xml: MulticastTTL	Clusters: <i>clustername</i> : Configuration: Multicast: Multicast TTL
weblogic.cluster.name	config.xml ClusterAddress	Clusters: <i>clustername</i> : Configuration: General: Cluster Address
weblogic.httpd.authRealmName	config.xml: AuthRealmName (WebAppComponent element)	Deployments: Web Applications: <i>applicationname</i> : Configuration: Other: Auth Realm Name
weblogic.httpd.charsets	config.xml: Charsets (WebServer element)	
weblogic.httpd.clustering.enable	config.xml: ClusteringEnabled (WebServer element)	
weblogic.httpd.defaultServerName	config.xml: DefaultServerName (WebServer element)	Servers: <i>servername</i> : Configuration: HTTP: Default Server Name

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.httpd.defaultServlet	web.xml: define a servlet-mapping with the URL pattern of / <servlet-mapping> element	
weblogic.httpd.defaultWebApp	config.xml: DefaultWebApp (WebServer element)	
weblogic.httpd.enable	config.xml: HttpdEnabled (Server element)	
weblogic.httpd.enableLogFile	config.xml: LoggingEnabled (WebServer element)	
weblogic.httpd.http.keepAliveSecs	config.xml: KeepAliveSecs (WebServer element)	
weblogic.httpd.https.keepAliveSecs	config.xml: HttpsKeepAliveSecs (WebServer element)	
weblogic.httpd.indexDirectories	config.xml: IndexDirectoryEnabled (WebAppComponent element)	Deployments: Web Applications: <i>applicationname</i> : Configuration: Files: Index Directories
weblogic.httpd.keepAlive.enable	config.xml: KeepAliveEnabled	Servers: <i>servername</i> : Configuration: HTTP: Enable Keep Alives
weblogic.httpd.logFileBufferKBytes	config.xml: LogFileBufferKBytes (WebServer element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.httpd.logFileFlushSecs	config.xml: LogFileFlushSecs (WebServer element)	
weblogic.httpd.logFileFormat	config.xml: LogFileFormat (WebServer element)	Services: Virtual Host: Log File Format
weblogic.httpd.logFileName	config.xml: LogFileName	Services: Virtual Host: Log File Name
weblogic.httpd.logRotationPeriod Mins	config.xml: LogRotationTimeBegin (WebServer element)	
weblogic.httpd.logRotationPeriod Mins	config.xml: LogRotationPeriodMin s (WebServer element)	
weblogic.httpd.logRotationType	config.xml: LogRotationType	Servers: <i>servername</i> : Logging: HTTP: Rotation Type
weblogic.httpd.maxLogFileSizeKBytes	config.xml: MaxLogFileSizeKBytes	Servers: <i>servername</i> : Logging: HTTP: Max Log File Size Kbytes
weblogic.httpd.mimeType	web.xml: mime-type <mime-mapping> element	
weblogic.httpd.postTimeoutSecs	config.xml: PostTimeoutSecs	Servers: <i>servername</i> : Configuration: HTTP: Post Timeout Secs
weblogic.httpd.servlet.extension CaseSensitive	config.xml: ServletExtensionCase Sensitive	Deployments: Web Applications: <i>applicationname</i> : Configuration: Files: Case Sensitive Extensions

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.httpd.servlet.reloadCheckSecs	config.xml: ServletReloadCheckSecs	Deployments: Web Applications: <i>applicationname</i> : Configuration: Files: Reload Period
weblogic.httpd.servlet.SingleThreadedModelPoolSize	config.xml: SingleThreadedServletPoolSize (WebAppComponent element)	Deployments: Web Applications: <i>applicationname</i> : Configuration: Files: Single Threaded Servlet Pool Size
weblogic.httpd.session.cacheEntries	weblogic.xml: CacheSize <param-name>/<param-value> element pair	Servers: <i>servername</i> : Configuration: SSL: Certificate Cache Size
weblogic.httpd.session.cookie.comment	weblogic.xml: CookieComment <param-name>/<param-value> element pair	
weblogic.httpd.session.cookie.domain	weblogic.xml: CookieDomain <param-name>/<param-value> element pair	
weblogic.httpd.session.cookie.maxAgeSecs	weblogic.xml: CookieMaxAgeSecs <param-name>/<param-value> element pair	
weblogic.httpd.session.cookie.name	weblogic.xml: CookieName <param-name>/<param-value> element pair	
weblogic.httpd.session.cookie.path	weblogic.xml: CookiePath <param-name>/<param-value> element pair	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.httpd.session.cookies.enable	weblogic.xml: CookiesEnabled <param-name>/<param-value> element pair	
weblogic.httpd.session.debug	weblogic.xml: SessionDebuggable<param-name>/<param-value> element pair	
weblogic.httpd.session.enable	weblogic.xml: TrackingEnabled <param-name>/<param-value> element pair	
weblogic.httpd.session.invalidat ionintervalSecs	weblogic.xml: InvalidationInterval Secs <param-name>/<param-value> element pair	
weblogic.httpd.session.jdbc.conn TimeoutSecs	weblogic.xml: JDBCConnectionTimeou tSecs <param-name>/<param-value> element pair	
weblogic.httpd.session.persisten tStoreDir	weblogic.xml: PersistentStoreDir <param-name>/<param-value> element pair	
weblogic.httpd.session.persisten tStorePool	weblogic.xml: PersistentStorePool <param-name>/<param-value> element pair	
weblogic.httpd.session.persisten tStoreShared	weblogic.xml: SessionPersistentSto reShared <param-name>/<param-value> element pair	

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.httpd.session.persistentStoreType	weblogic.xml: PersistentStoreType <param-name>/<param-value> element pair	
weblogic.httpd.session.sessionIDLength	weblogic.xml: IDLength <param-name>/<param-value> element pair	
weblogic.httpd.session.swapIntervalSecs	weblogic.xml: SwapIntervalSecs <param-name>/<param-value> element pair	
weblogic.httpd.session.timeoutSecs	weblogic.xml: TimeoutSecs<param-name>/<param-value> element pair	Servers: <i>servername</i> : Configuration: HTTP: Post Timeout Secs
weblogic.httpd.session.URLRewriting.enable	weblogic.xml: URLRewritingEnabled <param-name>/<param-value> element pair	
weblogic.httpd.tunneling.clientPingSecs	config.xml: TunnelingClientPingSecs	Servers: <i>servername</i> : Configuration: Tuning: Tunneling Client Ping
weblogic.httpd.tunneling.clientTimeoutSecs	config.xml: TunnelingClientTimeoutSecs (Server element)	Servers: <i>servername</i> : Configuration: Tuning: Tunneling Client Timeout
weblogic.httpd.tunnelingenabled	config.xml TunnelingEnabled (Server element)	Servers: <i>servername</i> : Configuration: Tuning: Enable Tunneling
weblogic.httpd.URLResource	config.xml: URLResource (WebServer element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.iiop.password	config.xml: DefaultIIOPPassword (Server element)	Servers: <i>servername</i> : Configuration: Protocols: Default IIOP Password
weblogic.iiop.user	config.xml: DefaultIIOPUser (Server element)	Servers: <i>servername</i> : Configuration: Protocols: Default IIOP User
weblogic.jdbc.connectionPool <ul style="list-style-type: none"> <li>■ url=<i>URL for JDBC Driver</i></li> <li>■ driver=<i>full package name for JDBC driver</i></li> <li>■ loginDelaySecs=<i>seconds between connections</i></li> <li>■ initialCapacity=<i>initial number of JDBC connections</i></li> <li>■ maxCapacity=<i>maximum number of JDBC connections</i></li> <li>■ capacityIncrement=<i>increment interval</i></li> <li>■ allowShrinking=<i>true to allow shrinking</i></li> <li>■ shrinkPeriodMins=<i>interval before shrinking</i></li> <li>■ testTable=<i>name of table for autorefresh test</i></li> <li>■ refreshTestMinutes=<i>interval for autorefresh test</i></li> <li>■ testConnsOnReserve=<i>true to test connection at reserve</i></li> <li>■ testConnsOnRelease=<i>true to test connection at release</i></li> <li>■ props=<i>props for JDBC connection</i></li> </ul>	config.xml JDBCConnectionPool Element <ul style="list-style-type: none"> <li>■ ConnLeakProfilingEnabled</li> <li>■ ACLName</li> <li>■ URL</li> <li>■ DriverName</li> <li>■ Properties</li> <li>■ LoginDelaySeconds</li> <li>■ InitialCapacity</li> <li>■ MaxCapacity</li> <li>■ CapacityIncrement</li> <li>■ CapacityEnabled</li> <li>■ ShrinkPeriodMinutes</li> <li>■ RefreshMinutes</li> <li>■ TestTableName</li> <li>■ TestConnectionsOnRelease</li> <li>■ SupportsLocalTransaction</li> <li>■ KeepLogicalConnOpenOnRelease</li> <li>■ Password</li> </ul>	
weblogic.jdbc.enableLogFile	config.xml: JDBCLoggingEnabled (Server element)	

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.jdbc.logFileName	config.xml: JDBCLogFileName (Server element)	
weblogic.jms.ConnectionConsumer	config.xml JMSConnectionConsumer element <ul style="list-style-type: none"> <li>■ MessagesMaximum</li> <li>■ Selector</li> <li>■ Destination</li> </ul>	
weblogic.jms.connectionFactoryArgs.<<factoryName>> <ul style="list-style-type: none"> <li>■ ClientID</li> <li>■ DeliveryMode</li> <li>■ TransactionTimeout</li> </ul>	config.xml: JMSConnectionFactory element <ul style="list-style-type: none"> <li>■ ClientID</li> <li>■ DefaultDeliveryMode</li> <li>■ TransactionTimeout</li> <li>■ UserTransactionsEnabled</li> <li>■ AllowCloseInOnMessage</li> </ul>	
weblogic.jms.connectionFactoryName	config.xml: JMSConnectionFactory element <ul style="list-style-type: none"> <li>■ JNDIName</li> </ul>	
weblogic.jms.connectionPool	ConnectionPool (JMSJDBCStore element)	
weblogic.jms.queue	config.xml: JNDIName StoreEnabled (JMSDestination element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.jms.queueSessionPool	config.xml: ConnectionConsumer ConnectionFactory ListenerClass AcknowledgeMode SessionsMaximum Transacted (JMSSessionPool element)	
weblogic.jms.tableNamePrefix	config.xml: PrefixName	
weblogic.jms.topic	config.xml JNDIName StoreEnabled (JMSTopic element)	Services: JMS: Connection Factories: JNDI Name
weblogic.jms.topicSessionPool	config.xml: ConnectionConsumer ConnectionFactory ListenerClass AcknowledgeMode SessionsMaximum Transacted (JMSSessionPool element)	
weblogic.jndi.transportableObjectFactories	config.xml: JNDITransportableObjectFactoryList (Server element)	Servers: <i>servername</i> :
weblogic.login.readTimeoutMillis SSL	config.xml LoginTimeoutMillis (SSL element)	Servers: <i>servername</i> :

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.security.audit.provider	config.xml AuditProviderClassName (Security element)	Security: General: Audit Provider Class
weblogic.security.certificate.authority	config.xml ServerCertificateChainFileName (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Server Certificate Chain File Name
weblogic.security.certificate.server	config.xml: ServerCertificateFileName (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Server Certificate File Name
weblogic.security.certificateCacheSize	config.xml: CertificateCacheSize (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Certificate Cache Size
weblogic.security.clientRootCA	config.xml: TrustedCAFileName (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Trusted CA File Name
weblogic.security.disableGuest	config.xml: GuestDisabled (Security element)	Security: General: Guest Disabled
weblogic.security.enforceClientCertificate	config.xml: ClientCertificateEnforced (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Client Certificate Enforced
weblogic.security.key.export.lifespan	config.xml: ExportKeyLifespan (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Export Key Lifespan
weblogic.security.key.server	config.xml: ServerKeyFileName (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Server Key File Name

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

---

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.security.ldaprealm.authentication	config.xml: AuthProtocol (LDAPRealm element)	
weblogic.security.ldaprealm.credential	config.xml: Credential (LDAPRealm element)	
weblogic.security.ldaprealm.factory	config.xml LdapProvider (LDAPRealm element)	
weblogic.security.ldaprealm.groupDN	config.xml: GroupDN (LDAPRealm element)	
weblogic.security.ldaprealm.groupIsContext	config.xml: GroupIsContext (LDAPRealm element)	
weblogic.security.ldaprealm.groupNameAttribute	config.xml: GroupNameAttribute (LDAP Realm element)	
weblogic.security.ldaprealm.groupUsernameAttribute	config.xml: GroupUsernameAttribute (LDAPRealm element)	
weblogic.security.ldaprealm.principal	config.xml: Principal (LDAPRealm element)	
weblogic.security.ldaprealm.ssl	config.xml: SSLEnable (LDAPRealm element)	
weblogic.security.ldaprealm.url	config.xml: LDAPURL (LDAPRealm element)	

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.security.ldaprealm.user Authentication	config.xml: UserAuthentication (LDAPRealm element)	
weblogic.security.ldaprealm.user DN	config.xml: UserDN (LDAPRealm element)	
weblogic.security.ldaprealm.user NameAttribute	config.xml: UserNameAttribute (LDAPRealm element)	
weblogic.security.ldaprealm.user PasswordAttribute	config.xml: UserPasswordAttribute (LDAPRealm element)	
weblogic.security.net.connection Filter	config.xml: ConnectionFilter (Security element)	
weblogic.security.ntrealm.domain	config.xml: PrimaryDomain (NTRealm element)	
weblogic.security.realm.cache.acl.enable	config.xml: ACLCacheEnable (CachingRealm element)	
weblogic.security.realm.cache.acl.size	config.xml: ACLCacheSize (CachingRealm element)	
weblogic.security.realm.cache.acl.ttl.negative	config.xml: ACLCacheTTLNegative (CachingRealm element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.security.realm.cache.acl.ttl.positive	config.xml: ACLCacheTTLPossible (CachingRealm element)	
weblogic.security.realm.cache.auth.enable	config.xml: AuthenticationCacheEnable (CachingRealm element)	
weblogic.security.realm.cache.auth.size	config.xml: AuthenticationCacheSize (CachingRealm element)	
weblogic.security.realm.cache.auth.ttl.negative	config.xml: AuthenticationCacheTTLNegative (CachingRealm element)	
weblogic.security.realm.cache.auth.ttl.positive	config.xml: AuthenticationCacheTTLPositive (CachingRealm element)	
weblogic.security.realm.cache.caseSensitive	config.xml: CacheCaseSensitive(CachingRealm element)	
weblogic.security.realm.cache.group.enable	config.xml: GroupCacheEnable (CachingRealm element)	
weblogic.security.realm.cache.group.size	config.xml: GroupCacheSize (CachingRealm element)	

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.security.realm.cache.group.ttl.negative	config.xml: GroupCacheTTLNegative (CachingRealm element)	
weblogic.security.realm.cache.group.ttl.positive	config.xml: GroupCacheTTLPositive (CachingRealm element)	
weblogic.security.realm.cache.permission.enable	config.xml: PermissionCacheEnable (CachingRealm element)	
weblogic.security.realm.cache.permission.size	config.xml: PermissionCacheSize (CachingRealm element)	
weblogic.security.realm.cache.permission.ttl.negative	config.xml: PermissionCacheTTLNegative (CachingRealm element)	
weblogic.security.realm.cache.permission.ttl.positive	config.xml: PermissionCacheTTLPositive (CachingRealm element)	
weblogic.security.realm.cache.user.enable	config.xml: UserCacheEnable (CachingRealm element)	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.security.realm.cache.user.size	config.xml: UserCacheSize (CachingRealm element)	
weblogic.security.realm.cache.user.ttl.negative	config.xml: UserCacheTTLNegative (CachingRealm element)	
weblogic.security.realm.cache.user.ttl.positive	config.xml: UserCacheTTLPositive (CachingRealm element)	
weblogic.security.realm.certAuthenticator	config.xml: CertAuthenticator (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Cert Authenticator
weblogic.security.SSL.ciphersuite	config.xml Ciphersuites (SSL element)	
weblogic.security.ssl.enable	config.xml: Enabled (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Enabled
weblogic.security.SSL.hostnameVerifier	config.xml HostnameVerifier (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Hostname Verifier
weblogic.security.SSL.ignoreHostNameVerification	config.xml HostNameVerification Ignored (SSL element)	
weblogic.security.SSLHandler.enable	config.xml: HandlerEnabled (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Handler Enabled

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.security.unixrealm.authProgram	config.xml: AuthProgram (UnixRealm element)	
weblogic.system.AdministrationPort	config.xml AdministrationPort (Server element)	Servers: <i>servername</i> : Configuration: General: Administration Port
weblogic.system.AdministrationPort	config.xml: AdministrationPort (Server element)	Servers: <i>servername</i> : Configuration: General: Administration Port
weblogic.system.bindAddr	config.xml: ListenAddress (Server element)	
weblogic.system.defaultProtocol	config.xml: DefaultProtocol (Server element)	Servers: <i>servername</i> : Configuration: Protocols: Default Protocol
weblogic.system.defaultSecureProtocol	config.xml: DefaultSecureProtocol (Server element)	Servers: <i>servername</i> : Configuration: Protocols: Default Secure Protocol
weblogic.system.enableConsole	config.xml: StdoutEnabled (Kernel element)	Servers: <i>servername</i> : Logging: General: Log to Stdout
weblogic.system.enableIIOP	config.xml: IIOPEnabled (Server element)	
weblogic.system.enableReverseDNSLookups	config.xml: ReverseDNSAllowed (Server element)	
weblogic.system.enableSetGID,	config.xml: PostBindGID	
weblogic.system.enableSetUID,	config.xml: PostBindUIDEnabled	

### 3 Migrating WebLogic Server 4.5 and 5.1 Applications to 6.x

<b>weblogic.properties file Property</b>	<b>.xml Configuration Attribute</b>	<b>Console Navigation</b>
weblogic.system.enableTGIOP	config.xml TGIOPEnabled (Server element)	Servers: <i>servername</i> :
weblogic.system.helpPageURL	config.xml HelpPageURL (Server element)	Servers: <i>servername</i> :
weblogic.system.home	config.xml: RootDirectory (Server element)	
weblogic.system.ListenPort	config.xml ListenPort (Server element)	Servers: <i>servername</i> : Configuration: SSL: Listen Port
weblogic.system.logFile	config.xml: FileName (Log element)	
weblogic.system.MagicThreadBackT oSocket	config.xml: MagicThreadDumpBackT oSocket (ServerDebug element)	
weblogic.system.MagicThreadDumpF ile	config.xml: MagicThreadDumpFile (ServerDebug element)	
weblogic.system.MagicThreadDumpH ost	config.xml: MagicThreadDumpHost (ServerDebug element)	
weblogic.system.magicThreadDumps	config.xml: MagicThreadDumpEnabl ed (ServerDebug element)	
weblogic.system.maxLogFileSize	config.xml: FileMinxSize (Log element)	

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.system.nativeIO.enable	config.xml: NativeIOEnabled (Server element)	Servers: <i>servername</i> : Configuration: Tuning: Enable Native IO
weblogic.system.nonPrivGroup	config.xml PostBindGID (UnixMachine element)	
weblogic.system.nonPrivUser	config.xml PostBindUID (UnixMachine element)	
weblogic.system.percentSocketReaders	config.xml: ThreadPoolPercentSocketReaders (Kernel element)	Servers: <i>servername</i> : Configuration: Tuning: Socket Readers
weblogic.system.readTimeoutMillis	config.xml LoginTimeoutMillis (Server element)	Servers: <i>servername</i> :
weblogic.system.SSL.useJava	config.xml: UseJava (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Use Java
weblogic.system.SSLListenPort	config.xml: ListenPort (SSL element)	Servers: <i>servername</i> : Configuration: SSL: Listen Port
weblogic.system.startupFailureIsFatal	config.xml FailureIsFatal (StartupClass element)	
weblogic.system.user	config.xml: SystemUser (Security element)	
weblogic.system.weight	config.xml ClusterWeight (Server element)	Servers: <i>servername</i> : Configuration: Cluster: Cluster Weight

weblogic.properties file Property	.xml Configuration Attribute	Console Navigation
weblogic.workspace.showUserKeysOnly	config.xml: WorkspaceShowUserKeysOnly (Server element)	Servers: <i>servername</i> : Configuration: Tuning: Show Only Workspace User Keys
weblogic.zac.enable	config.xml: ZACEnabled (Server element)	Servers: <i>servername</i>
weblogic.zac.publishRootProp	config.xml: ZACPublishRoot (Server element)	Servers: <i>servername</i>

## Migrating Web Applications

In order to migrate applications into a Web Application deployed on WebLogic Server 6.0 or 6.1, the applications' files must be placed within a directory structure that follows a specific pattern. For development, these files can be left in an exploded directory format. However, for production situations, it is highly recommended that you bundle your applications into a .war file as a single Web Application. For more information on Web Applications see [Understanding WebLogic Server Applications at http://e-docs.bea.com/wls/docs61/programming/concepts.html](http://e-docs.bea.com/wls/docs61/programming/concepts.html) and [Assembling and Configuring Web Applications at http://e-docs.bea.com/wls/docs61/webapp/index.html](http://e-docs.bea.com/wls/docs61/webapp/index.html).

The following sections provide information you need to know about migrating and deploying Web Applications, including a procedure for migrating a simple servlet from WebLogic Server 5.1 to WebLogic Server 6.1:

- “Web Applications Directory Structure” on page 3-25
- “XML Deployment Descriptors” on page 3-26
- “WAR Files” on page 3-27
- “Deploying Web Applications” on page 3-27

- “Session Migration” on page 3-28
- “JavaServer Pages (JSPs) and Servlets” on page 3-28
- “Migrating a Simple Servlet from WebLogic Server 5.1 to WebLogic Server 6.1” on page 3-29

## Web Applications Directory Structure

Web Applications are organized in a specified directory structure so that they can be archived and deployed on WebLogic Server. All servlets, classes, static files, and other resources belonging to a Web Application are organized under a directory hierarchy. The root of this hierarchy defines the document root of your Web Application. All files under this root directory can be served to the client, except for files under the special directories `WEB-INF` and `META-INF` located in the root directory. The root directory should be named with the name of your Web Application and placed inside the `wlserver6.1/config/domainName/applications` directory.

The following diagram illustrates the directory structure of any Web Application.

```

WebApplicationRoot/(Publically available files such as
| .jsp, .html, .jpg, .gif)
|
+WEB-INF/--+
|
+ classes/(directory containing
|           Java classes including
|           servlets used by the
|           Web Application)
|
+ lib/(directory containing
|      jar files used by the
|      Web Application)
|
+ web.xml
|
+ weblogic.xml

```

When you convert your `weblogic.properties` file, the appropriate `web.xml` and `weblogic.xml` files are created for you under the directory `wlserver6.1/config/domainName/applications/DefaultWebApp_myserver/WEB-INF`. Follow the preceding directory structure and place the `.xml` files in the `wlserver6.1/config/domainName/applications/webAppName/WEB-INF`

directory that you create. All applications must be placed inside a `wlserver6.1/config/domainName/applications` directory in order to be deployed. For more information, see [Developing WebLogic Server Applications at `http://e-docs.bea.com/wls/docs61/programming/index.html`](http://e-docs.bea.com/wls/docs61/programming/index.html).

## XML Deployment Descriptors

The Web Application Deployment Descriptor (`web.xml`) is a standard J2EE descriptor used to register your servlets, define servlet initialization parameters, register JSP tag libraries, define security constraints, and other Web Application parameters. For detailed instructions on creating the deployment descriptor, see [Writing the Web Application Deployment Descriptor at `http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#web-xml`](http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#web-xml).

There is also a WebLogic-specific Deployment Descriptor (`weblogic.xml`). In this file you define JSP properties, JNDI mappings, security role mappings, and HTTP session parameters. The WebLogic-specific deployment descriptor also defines how named resources in the `web.xml` file are mapped to resources residing elsewhere in WebLogic Server. For detailed instructions on creating the WebLogic-specific deployment descriptor, see [Writing the WebLogic-specific Deployment Descriptor at `http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#weblogic-xml`](http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#weblogic-xml). This file may not be required if you have no need for any of the above mentioned properties, mappings, or parameters.

Use the `web.xml` and `weblogic.xml` files, in conjunction with the console, to configure your applications. The `.xml` files can be viewed through any text editor. To edit them, simply make your changes and save the file as `web.xml` or `weblogic.xml` with the appropriate path as specified by the prescribed directory structure. See [Assembling and Configuring Web Applications at `http://e-docs.bea.com/wls/docs61/webapp/index.html`](http://e-docs.bea.com/wls/docs61/webapp/index.html) for more information. If you do not want to deploy your applications together as a single Web Application, you need to split up the `.xml` files that have been created for you, creating the appropriate `.xml` files specific to each Web Application.

## WAR Files

A `.war` file is a Web Application archive. If you have correctly followed the prescribed directory structure of a Web Application and created the appropriate `web.xml` and `weblogic.xml` files, it is strongly recommended that in production environments your applications be bundled together in a Web Application deployed as a `.war` file. Once you have bundled your applications into a `.war` file, it is important to remove the previously existing directory structure so that WebLogic Server only has one instance of each application.

Use the following command line from the root directory containing your Web Application to create a `.war` file, replacing '`webAppName`' with the specific name you have chosen for your Web Application:

```
jar cvf webAppName.war *
```

You now have created a `.war` file that contains all the files and configuration information for your Web Application.

## Deploying Web Applications

To deploy your bundled Web Applications properly, place the appropriate `.war` file in the `c:\wlserver6.1/config/domainName/applications` directory. You can also install the application through the Administration Console. To do so, go to the console home and choose Install Applications under the Getting Started menu. Select the correct `.war` file and it will be installed automatically. Note that it is necessary to have your applications reside in a

`c:\wlserver6.1/config/domainName/applications` directory in order for them to work.

Web Applications should be deployed automatically after they have been installed. Check to see that they are deployed under the Deployments node in the left hand pane of the Administration Console.

You can configure certain deployment attributes for your Web Application using the Administration Console. Select the Web Applications node under the Deployments heading. Select your Web Application. Click on the appropriate tab to configure. For more information on setting attributes in the console, see the Web Application section of the [Console Help](#) at

<http://e-docs.bea.com/wls/docs61/ConsoleHelp/webappcomponent.html>.

## Session Migration

WebLogic Server 6.0 and later does not recognize cookies from previous versions because cookie format changed with 6.0. WebLogic Server will ignore cookies with the old format, and create new sessions.

The default name for cookies has changed from 5.1, when it was `WebLogicSession`. Beginning in WebLogic 6.0, cookies are named `JSESSIONID` by default.

See *[weblogic.xml Deployment Descriptor Elements](#)* at

[http://e-docs.bea.com/wls/docs61/webapp/weblogic\\_xml.html](http://e-docs.bea.com/wls/docs61/webapp/weblogic_xml.html) for more information.

## JavaServer Pages (JSPs) and Servlets

This section contains information specific to JSPs and servlets that may be pertinent to your applications.

- Some changes will be necessary in code (both Java and HTML) where the code refers to URLs which may be different when servlets and JSPs are deployed in a Web Application other than the default Web Application. See the *[Servlet Documentation](#)* at <http://e-docs.bea.com/wls/docs61/servlet/admin.html#servlet-urls> for more information. If relative URLs are used and all components are contained in the same Web Application, these changes are not necessary.
- Only serializable objects may be stored in a session if your application is intended to be distributable.
- You must convert your `weblogic.properties` to XML attributes in `web.xml` and/or `weblogic.xml`. For additional information on this process, see the conversion section of the *[Console Help](#)* at <http://e-docs.bea.com/wls/docs61/ConsoleHelp/conversion.html>.
- ACLs are now defined as security constraints in `web.xml`.
- Server-side-includes are not supported. You must use JSP to achieve this functionality.

- This version of WebLogic Server is now fully compliant with the Servlet 2.3 specification.

## Migrating a Simple Servlet from WebLogic Server 5.1 to WebLogic Server 6.1

The following procedure migrates the simple Hello World Servlet that was provided with WebLogic 5.1 Server to WebLogic Server 6.1.

1. Create the correct directory structure, as described in *Programming HTTP Servlets* at <http://e-docs.bea.com/wls/docs61/servlet/admin.html>. This involves creating a root application directory, such as C:/hello, as well as a C:/hello/WEB-INF directory and a C:/hello/WEB-INF/classes directory. Place the HelloWorldServlet.java file inside the C:/hello/WEB-INF/classes directory.
2. Create a web.xml file for this servlet. If you have converted your weblogic.properties file, a web.xml file has already been created for you. If you registered HelloWorldServlet in your weblogic.properties file before you converted it, the servlet will be properly configured in your new web.xml file. An .xml file can be created with any text editor. The following is an example of a basic web.xml file that could be used with the HelloWorldServlet.

```
<!DOCTYPE web-app (View Source for full doctype...)>
- <web-app>
- <servlet>
<servlet-name>HelloWorldServlet</servlet-name>
<servlet-class>examples.servlets.HelloWorldServlet</servlet-cla
ss>
</servlet>
- <servlet-mapping>
<servlet-name>HelloWorldServlet</servlet-name>
<url-pattern>/hello/*</url-pattern>
</servlet-mapping>
</web-app>
```

For more information on web.xml files, see *Writing the Web Application Deployment Descriptor* at <http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#web-xml>. A weblogic.xml file is not necessary with such a simple, stand-alone servlet as HelloWorld.

For more information on `weblogic.xml` files, see [Writing the WebLogic-Specific Deployment Descriptor](http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#eblogic-xml) at <http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html#eblogic-xml>.

3. Move the `web.xml` file from `wlserver6.1/config/domainName/applications/DefaultWebApp_myserver/WEB-INF` to `C:/hello/WEB-INF/`.
4. Set up your development environment (see [Establishing a Development Environment](http://e-docs.bea.com/wls/docs61/programming/environment.html#denv) at <http://e-docs.bea.com/wls/docs61/programming/environment.html#denv> for more information) and compile the `HelloWorldServlet` with a command like the following:

```
C:\hello\WEB-INF\classes>javac -d . HelloWorldServlet.java
```

This should compile the file and create the correct package structure.

5. The servlet can now be bundled into an archive `.war` file with the following command:

```
jar cvf hello.war *
```

This command will create a `hello.war` file and place it inside the `C:/hello` directory.

6. To install this Web Application, start your server and open the Administration Console. Under the Getting Started menu, choose Install Applications. Browse to the newly created `.war` file and click Upload.

The servlet should now be deployed and appear under the Web Applications node under Deployments, in the left-hand pane of the console.

7. To call the servlet, type the following in your browser URL window:  
`http://localhost:7001/hello/hello`.

In this case `/hello/` is the context path of the servlet. This is determined by the naming of the `.war` file, in this case `hello.war`. The second `/hello` was mapped in the servlet mapping tags inside the `web.xml` file.

# Migrating and Converting Enterprise JavaBeans Applications

The following sections describe Enterprise Java Beans migration and conversion procedures.

## EJB Migration Considerations

Consider the following when migrating Enterprise JavaBeans to WebLogic Server 6.x.

- WebLogic Server Version 6.0 and 6.1 both support the Enterprise JavaBeans 1.1 and 2.0 specifications.
- The XML parser is stricter with XML deployment descriptors in WebLogic 6.0 and 6.1 than it was in WebLogic 5.1. Some errors allowed in earlier versions are no longer permitted. This is described in *Programming WebLogic EJB* at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_whatsnew.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_whatsnew.html).
- EJB 1.1 beans are deployable in WebLogic Server 6.0 and 6.1. However, if you are developing new beans, it is recommended that you use EJB 2.0. EJB 1.1 beans can be converted to 2.0 using the DDConverter utility. For more information see the *DDConverter* documentation at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_utilities.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_utilities.html).
- You can upgrade EJB 1.0 deployment descriptors to EJB 2.0 using the DDConverter utility, but first those descriptors must be upgraded to 1.1. WebLogic Server 5.1 deployment descriptors can be upgraded to 6.0 or 6.1 to take advantage of new features in WebLogic Server 6.0 and 6.1. Details on the DDConverter utility are provided in *Programming WebLogic EJB* at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_utilities.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_utilities.html).
- The finder expressions feature of EJB 1.1 is no longer supported. This is the only non-supported feature of EJB 1.1.
- Deploying beans is described in *Programming WebLogic EJB* at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_deployover.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_deployover.html).

- If `ejbc` has not been run on an EJB, WebLogic Server 6.1 will run `ejbc` automatically when the bean is deployed. You do not need to compile beans with `ejbc` before deploying. If you wish to run `ejbc` during startup, you may do so. See details in *Programming WebLogic EJB* at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_deployover.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_deployover.html).
- An EJB deployment includes a standard deployment descriptor in the `ejb-jar.xml` file. The `ejb-jar.xml` must conform to either the EJB 1.1 DTD (document type definition) or the EJB 2.0 DTD.
- An EJB deployment needs the `weblogic-ejb-jar.xml` file, a WebLogic Server-specific deployment descriptor that includes configuration information for the WebLogic Server EJB container. This file must conform to the WebLogic Server 5.1 DTD or the WebLogic Server 6.0 DTD.
- In order to specify the mappings to the database, Container-Managed Persistence Entity Beans require a CMP deployment descriptor which conforms to either the WebLogic Server 5.1 CMP DTD, the WebLogic Server 6.0 EJB 1.1 DTD, or the WebLogic Server 6.0 EJB 2.0 DTD.

## EJB Migration Recommendations

- Use `TxDataSource`

EJBs should always get their database connections from a `TxDataSource`. This allows the EJB container's transaction management to interface with the JDBC connection, and it also supports XA transactions.

The WebLogic Server 6.x CMP Deployment Descriptor supports `TxDatasources` and should be used instead of the WebLogic Server 5.1 CMP Deployment Descriptor which only specifies a connection pool.
- Use a fast compiler with `ejbc`

The WebLogic Server EJB Compiler (`weblogic.ejbc`) generates java code that is then compiled by the java compiler. By default, WebLogic Server uses the `javac` compiler included with the bundled JDK. The EJB Compiler runs much faster when a faster java compiler is used. Use the `-compiler` option to specify an alternate compiler as follows:

```
java weblogic.ejbc -compiler sj pre_AccountEJB.jar
AccountEJB.jar
```

- Correct errors before deploying the EJB on the WebLogic Server 6.1

The WebLogic Server 6.1 EJB Compiler (ejbc) includes additional verification that was missing from earlier WebLogic Server releases. It is possible that an EJB deployed in a previous WebLogic Server version without error, but WebLogic Server 6.1 finds and complains about the error. These errors must be corrected before the EJB is deployed in WebLogic Server 6.1.

For instance, WebLogic Server 6.1 ensures that a method exists if a transaction attribute is set for that method name. This helps identify a common set of errors where transaction attributes were mistakenly set on non-existent methods.

- Look at the WebLogic Server 6.1 examples

The WebLogic Server distribution contains several EJB examples including their deployment descriptors. The EJB examples can be found in the `samples/examples/ejb` and `samples/examples/ejb20` directories of the WebLogic Server 6.1 distribution.

The following table shows the descriptor combinations supported by WebLogic Server 6.1.

EJB Version	WebLogic Server Version	CMP Version
Any existing WebLogic Server 5.1 deployment uses the following combination and can be deployed without changing descriptors or code in WebLogic Server 6.1.		
1.1	5.1	5.1
The following combination uses a WebLogic Server 6.0 deployment descriptor so that new WebLogic Server 6.x features can be specified in the <code>weblogic-ejb-jar.xml</code> file.		
1.1	6.x	5.1
The below combinations include a WebLogic Server 6.0 CMP deployment descriptor. The WebLogic Server 6.x EJB 1.1 CMP deployment descriptor allows multiple EJBs to be specified within a single EJB JAR file, and it supports using a <code>TxDataSource</code> which is required when an EJB is enlisted in a two-phase / XA transaction.		
1.1	5.1	6.x
1.1	6.x	6.x
EJB 2.0 beans always use the WebLogic Server 6.x deployment descriptors.		
2.0	6.x	6.x

For more information on Enterprise JavaBeans, see *Enterprise JavaBeans Components* and *Programming WebLogic EJB* at <http://e-docs.bea.com/wls/docs61/ejb/index.html>.

# Steps for Migrating a 1.0 EJB from WebLogic Server 4.5.x to WebLogic Server 6.1

WebLogic Server 3.1.x, 4.0.x, and 4.5.x supported the EJB 1.0 specification. To migrate a 1.0 EJB from WebLogic Server 4.5 to WebLogic Server 6.1:

1. Convert the EJB 1.0 deployment descriptor to either the EJB 1.1 or the EJB 2.0 XML deployment descriptor. You can do this automatically using the [DDCreator](#) tool.
2. Package the deployment in a JAR file.
3. Run the WebLogic Server EJB compiler (ejbc) to compile the JAR file. The ejbc tool ensures that when the EJB compiles, it conforms to either the EJB 1.1 or EJB 2.0 specifications.
4. Correct any compliance errors before deploying the EJB in the EJB container.

To ensure EJB 1.1 or 2.0 compliance, make the following changes to the EJB 1.0 beans:

- EJB 1.0 beans referred to the `SessionContext` or `EntityContext` transient. When deploying EJB 1.1 or 2.0 beans, the reference cannot be transient. For example:

```
private transient SessionContext ctx;
```

should be:

```
private SessionContext ctx;
```

- The `ejbCreate` method for EJB 1.0 CMP entity beans had a void return type. When deploying EJB 1.1 or 2.0 beans, the return type must be the primary key class which allows you to write a bean-managed persistent entity bean and then sub-class it with a CMP implementation. For example:

```
public void ejbCreate (String name) {  
    firstName = name;  
}
```

should be:

```
public AccountPK ejbCreate (String name) {
    firstName = name;
    return null; // required by the EJB specification
}
```

- In EJB 1.1 or 2.0, entity beans cannot use bean-managed transactions. Instead, they must run with a container-managed transaction attribute (for example, Required, Mandatory, etc.).

## Steps for Migrating a 1.1 EJB Bean from WebLogic Server 5.1 to WebLogic Server 6.1

The WebLogic Server 5.1 deployment descriptor only allows the exclusive or read-only concurrency options. The database concurrency option is available when upgrading to the WebLogic Server 6.x `weblogic-ejb-jar.xml` file. For more information about this option, see information on database concurrency in [weblogic-ejb-jar.xml Deployment Descriptors](#)

The WebLogic Server 6.x CMP deployment descriptor allows multiple EJBs to be specified and it supports using a `TxDatasource` instead of a connection pool. Using a `TxDatasource` is required when XA is being used with EJB 1.1 CMP.

To migrate a 1.1 EJB from WebLogic Server 5.1 to WebLogic Server 6.1:

1. Open the Administration Console. From the home page, click on Install Applications under the Getting Started heading.
2. Locate the JAR file you wish to migrate by clicking the Browse button, then click Open and then Upload. Your bean should now be automatically deployed.
3. Run a `setEnv` script in a client window and set your development environment. (For more information, see [Establishing a Development Environment at `http://e-docs.bea.com/wls/docs61/programming/environment.html#denv`](http://e-docs.bea.com/wls/docs61/programming/environment.html#denv).)
4. Compile all the needed client classes. For example, using the Stateless Session Bean sample that was provided with WebLogic Server 5.1, you would use the following command:

```
javac -d %CLIENTCLASSES% Trader.java TraderHome.java  
TradeResult.java Client.java
```

5. To run the client, enter this command:

```
java examples.ejb.basic.statelessSession.Client
```

## Steps for Converting an EJB 1.1 to an EJB 2.0

To convert an EJB 1.1 bean to a EJB 2.0 bean, you can use the WebLogic Server *DDConverter* utility.

We recommend that you develop EJB 2.0 beans in conjunction with WebLogic Server 6.1. However, for 1.1 beans already used in production, it is not necessary to convert them to 2.0 beans. EJB 1.1 beans are deployable with WebLogic Server 6.1. If you do wish to convert 1.1 beans to 2.0 beans, see the *DDConverter* documentation at [http://e-docs.bea.com/wls/docs61/ejb/EJB\\_utilities.html](http://e-docs.bea.com/wls/docs61/ejb/EJB_utilities.html) for more information.

The basic steps required to convert a simple CMP 1.1 bean to a 2.0 bean are as follows:

1. Make the bean class abstract. EJB 1.1 beans declare CMP fields in the bean. CMP 2.0 beans use abstract `getXXX` and `setXXX` methods for each field. For instance, 1.1 Beans will use `public String name`. 2.0 Beans should use `public abstract String getName()` and `public abstract void setName(String n)`. With this modification, the bean class should now read the container-managed fields with the `getName` method and update them with the `setName` method.
2. Any CMP 1.1 finder which used `java.util.Enumeration` should now use `java.util.Collection`. Change your code to reflect this. CMP 2.0 finders cannot return `java.util.Enumeration`.

## Migrating EJBs from other J2EE Application Servers

Any EJB that complies with the EJB 1.1 or EJB 2.0 specifications may be deployed in the WebLogic Server 6.1 EJB container. Each EJB JAR file requires an `ejb-jar.xml` file, a `weblogic-ejb-jar.xml` deployment descriptor and a CMP deployment

descriptor if CMP entity beans are used. The WebLogic Server EJB examples located in `samples/examples/ejb` and `samples/examples/ejb20` of the WebLogic Server distribution include sample weblogic deployment descriptors.

## Creating an Enterprise Application

An Enterprise Application consists of assembled components, and is a `.jar` file with an `.ear` extension. An `.ear` file contains all of the `.jar` and `.war` component archive files for an application and an XML descriptor that describes the bundled components. The `META-INF/application.xml` deployment descriptor contains an entry for each Web and EJB module, and additional entries to describe security roles and application resources such as databases.

```
EnterpriseApplicationStagingDirectory/
|
+ .jar files
|
+ .war files
|
+META-INF/-+
|
+ application.xml
```

To create an `.ear` file:

1. Assemble all of the Web Archive and EJB Archives for your application and place them under your staging directory.
2. Copy the `.war` and EJB `.jar` files into the staging directory and then create a `META-INF/application.xml` deployment descriptor for the application. Follow the directory structure depicted above.

The `application.xml` file contains a descriptor for each component in the application, using a DTD supplied by Sun Microsystems. For more information on the `application.xml` file, see [Client Application Deployment Descriptor Elements at](#)

[http://e-docs.bea.com/wls/docs61/programming/app\\_xml.html](http://e-docs.bea.com/wls/docs61/programming/app_xml.html). Note that if you are using JSPs and want them to compile at run time you must have

the home and remote interfaces of the bean included in the classes directory of your .war file.

3. Create the Enterprise Archive by executing a jar command like the following in the staging directory:

```
jar cvf myApp.ear *
```

4. Click on the Install Applications link under the Getting Started heading in the home page of the console and place the .ear file in the `/wlserver6.1/config/domainName/applications` directory. For more information on Enterprise Applications, see [Staging Enterprise Applications at http://e-docs.bea.com/wls/docs61/programming/packaging.html](http://e-docs.bea.com/wls/docs61/programming/packaging.html).

## Understanding J2EE Client Applications

WebLogic Server supports J2EE client applications, packaged in a jar file with a standard XML deployment descriptor. Client applications in this context are clients that are not Web browsers. They are Java classes that connect to WebLogic Server using Remote Method Invocation (RMI). A Java client can access Enterprise JavaBeans, JDBC connections, JMS messaging, and other services using RMI. Client applications range from simple command line utilities that use standard I/O to highly interactive GUI applications built using the Java Swing/AWT classes.

To execute a WebLogic Server Java client, the client computer needs the `weblogic_sp.jar` file, the `weblogic.jar` file, the remote interfaces for any RMI classes and Enterprise Beans that are on WebLogic Server, as well as the client application classes. To simplify maintenance and deployment, it is a good idea to package a client-side application in a jar file that can be added to the client's classpath along with the `weblogic.jar` and `weblogic_sp.jar` files. The `weblogic.ClientDeployer` command line utility is executed on the client computer to package a client application to this specification. For more information about J2EE client applications, see [Staging and Deploying Client Applications at http://e-docs.bea.com/wls/docs61/programming/packaging.html](http://e-docs.bea.com/wls/docs61/programming/packaging.html).

# Migrating JMS Applications

WebLogic Server 6.1 supports the *JavaSoft JMS specification version 1.0.2* at <http://www.javasoft.com/products/jms/docs.html>.

In order to use your existing JMS applications, you must first confirm your version of WebLogic Server, and then perform the appropriate migration procedures provided in the “*Migrating WebLogic JMS Application*” chapter in *Programming WebLogic JMS*.

Before beginning the migration procedure, you should check the following list to confirm whether migration is supported for your version of WebLogic JMS, and to find out whether special migration rules apply to that release:

- Weblogic Server 4.5.1 — Migration is supported *only* for SP14. Customers running all service packs should contact BEA Support.
- Weblogic Server 5.1 — Customers running SP07 or SP08 should contact BEA Support before migrating existing JDBC stores to versions 6.0 or 6.1.
  - In order to migrate object messages, the object classes need to be in the Weblogic Server 6.0 or higher server classpath.
  - For destinations that are not configured in Weblogic Server 6.0 or higher, the migrated messages will be dropped and the event will be logged.

**Note:** WebLogic Events are being deprecated and being replaced by JMS messages with NO\_ACKNOWLEDGE or MULTICAST\_NO\_ACKNOWLEDGE delivery modes. Each of these delivery modes is described in *Programming WebLogic JMS* at <http://e-docs.bea.com/wls/docs61/jms/fund.html>.

# Additional Migration and Deployment Considerations

The following sections provide additional information that may be useful when you deploy applications on WebLogic Server 6.x. Deprecated features, upgrades, and the important changes that have been made in WebLogic Server 6.1 are noted.

- “Standalone HTML and JSPs” on page 3-40
- “Applications and Managed Servers” on page 3-41
- “Java Transaction API (JTA)” on page 3-41
- “Java Database Connectivity (JDBC)” on page 3-41
- “RMI” on page 3-42
- “Internationalization (I18N)” on page 3-43
- “Security” on page 3-44
- “WAP Applications” on page 3-45
- “Session Migration” on page 3-45
- “Web Components” on page 3-46
- “XML 6.0->6.1 Parser” on page 3-47
- “Deprecated APIs and Features” on page 3-47
- “Removed APIs and Features” on page 3-48

## Standalone HTML and JSPs

In the original domain provided with WebLogic Server 6.1, as well as in any domains that have been created using the `weblogic.properties` file converter, `wlserver6.1/config/domainName/applications/DefaultWebApp_myserver` directory is created. This directory contains files made available by your Web server.

You can place HTML and JSP files here and make them available, separate from any applications you install. If necessary, you can create subdirectories within the `DefaultWebApp_myserver` directory to handle relative links, such as image files.

## Applications and Managed Servers

By default, applications are deployed on the Administration Server's `config.xml` block and JVM. However, in most cases, this is not good practice. You should use the Administration Server only for administrative purposes, and define new managed servers and associate the applications with those servers using the Administration Console. For more information, see *Configuring WebLogic Servers and Clusters at <http://e-docs.bea.com/wls/docs61/adminguide/config.html>* and *Domains, the Administration Server, and Managed Servers at <http://e-docs.bea.com/wls/docs61/adminguide/overview.html>*.

## Java Transaction API (JTA)

The following changes in JTA have taken place:

- WebLogic Server 6.1 supports the JTA 1.0.1 specification. Updated JTA documentation is provided in *Programming WebLogic JTA at <http://e-docs.bea.com/wls/docs61/jta/index.html>*.
- Based on the inclusion of support for JTA, the JTS JDBC driver (with properties in `weblogic.jts.*` and URL `jdbc:weblogic:jts:..`) has been replaced by a JTA JDBC/XA driver. Existing properties are available for backward compatibility, but you should change the class name and properties to reflect the JTS to JTA name change.

## Java Database Connectivity (JDBC)

The following changes have been made in JDBC:

- The WebLogic T3 API is deprecated in WebLogic Server 6.0 and 6.1; use the RMI JDBC driver in its place. This also applies to users migrating from WebLogic Server 4.5.x.

- The `weblogic.jdbc20.*` packages are being replaced with `weblogic.jdbc.*` packages. All WebLogic JDBC drivers are now compliant with JDBC 2.0.
- If you have a current connection and are using a `preparedStatement`, and the stored procedure gets dropped in the DBMS, use a new name to create the stored procedure. If you recreate the stored procedure with the same name, the `preparedStatement` will not know how to access the newly created stored procedure—it is essentially a different object with the same name.

## RMI

The following tips are for users migrating to WebLogic Server 6.1 who used RMI in their previous version of WebLogic Server:

- Re-run the WebLogic RMI compiler, `weblogic.rmic`, on any existing code to regenerate the wrapper classes so they are compatible with WebLogic Server 6.1.
- Use `java.rmi.Remote` to tag interfaces as remote. Do not use `weblogic.rmi.Remote`.
- Use `java.rmi.*Exception` (e.g., `import java.rmiRemoteException;`). Do not use `weblogic.rmi.*Exception`.
- Use JNDI instead of `*.rmi.Naming`.
- Use `weblogic.rmic` to generate dynamic proxies and bytecode; with the exception of RMI IIOP, stubs and skeletons classes are no longer generated.

**Note:** For more information, see [Programming WebLogic RMI](#).

- Use `weblogic.rmi.server.UnicastRemoteObject.exportObject()` to get a stub instance.
- The RMI examples have not currently been updated to use `java.rmi.*` and JNDI. The examples will be revised to reflect `java.rmi.*` and JNDI in a future release.

## FileServlet

In WebLogic Server 6.1 Service Pack 2 and later, the behavior of FileServlet, which is the default servlet for a Web Application, has changed. FileServlet now includes the `SERVLET_PATH` when determining the source filename. This means that it is possible to explicitly only serve files from specific directories by mapping the FileServlet to `/dir/*` etc.

See [Setting Up a Default Servlet](#).

## Internationalization (I18N)

Several internationalization and localization changes have been made in this version:

- Changes have been made to the log file format that affect the way that messages are localized. The new message format also has additions to the first line: *begin marker, machine name, server name, thread id, user id, tran id, and message id*.
- There is a new internationalized logging API through which users can log messages in the server and clients.
- Clients log to their own logfiles which are in the same format as the server logfiles, with the exception of the *servername* and *threadid* fields.
- `LogServicesDef` is deprecated. Instead, use the internationalized API or `weblogic.logging.NonCatalogLogger` (when internationalization is not required).

For details on internationalization in this version, see [BEA WebLogic Server Internationalization Guide](#) at

<http://e-docs.bea.com/wls/docs61/i18n/index.html>.

## Security

It is highly recommended that users read our documentation on *Programming WebLogic Security* at <http://e-docs.bea.com/wls/docs61/security/index.html> for the most detailed information on using security with WebLogic Server 6.1. WebLogic Server users migrating from earlier releases should read the following list of tips and issues:

- A large portion of implementing security in the WebLogic Server environment is configuration. To migrate a current security configuration, convert the `weblogic.properties` file to XML attributes in `config.xml` using the Administration Console. Details on converting `weblogic.properties` files are described in the *Console Help* documentation at <http://e-docs.bea.com/wls/docs61/ConsoleHelp/conversion.html>. For a complete mapping of the security properties to XML attributes, see “Managing Security” in the *Administration Guide* at <http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html>.
- The name of the default security realm changed from `WLPropertyRealm` to the `File` realm. Realm attributes are now stored in the `fileRealm.properties` file instead of the `weblogic.properties` file.
- At some point, redefine your realm and authorization attributes through the Administration Console. The resulting information is stored in the `fileRealm.properties` file. For information about creating a custom realm in WebLogic Server 6.x, see [Writing a Custom Security Realm](#).
- It is highly recommended that at the end of installation, you check all security settings to make sure they are the appropriate ones for their environment.
- ACLs can no longer be used to specify security for stand-alone servlets because stand-alone servlets have been completely replaced by web applications. Web applications can only be secured using the web app's deployment descriptors as defined in the Servlet 2.3 specification.

## WAP Applications

To run a WAP application on WebLogic Server 6.1, you must now specify the MIME types associated with WAP in the `web.xml` file of the web application. In WebLogic Server 5.1, the default mime-type can be set using `weblogic.httpd.defaultMimeType` in `weblogic.properties` where its default value is "text/plain". WebLogic Server 6.0, WebLogic Server 6.1, and WebLogic Server 7.0 do not have a default `mime-type`. You must explicitly specify `mime-type` for each extension in the `web.xml` file. For information on required MIME types see [Programming WebLogic Server for Wireless Services at `http://e-docs.bea.com/wls/docs61/wireless/index.html`](http://e-docs.bea.com/wls/docs61/wireless/index.html). For information on creating and editing a `web.xml` file, see [Writing Web Application Deployment Descriptors at `http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html`](http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html).

An example configuration of the `mime-types` in the `web.xml` file:

```
<web-app>
  <mime-mapping>
    <extension>tiff</extension>
    <mime-type>image/tiff</extension>
  </mime-mapping>
  <mime-mapping>
    <extension>tif</extension>
    <mime-type>image/tiff</extension>
  </mime-mapping>
</web-app>
```

## Session Migration

WebLogic Server 6.0 and later does not recognize cookies from previous versions because cookie format changed with 6.0. WebLogic Server ignores cookies with the old format, and creates new sessions.

The default name for cookies has changed from 5.1, when it was `WebLogicSession`. Beginning in WebLogic 6.0, cookies are named `JSESSIONID` by default.

See [weblogic.xml Deployment Descriptor Elements](#) at [http://e-docs.bea.com/wls/docs61/webapp/weblogic\\_xml.html](http://e-docs.bea.com/wls/docs61/webapp/weblogic_xml.html) for more information.

## Web Components

The following tips are for users migrating to WebLogic Server 6.1 who used Web Components in their previous version of WebLogic Server:

- All Web components in WebLogic Server now use Web Applications as the mechanism for defining how WebLogic Server serves up JSPs, servlets, and static HTML pages. In a new installation of WebLogic Server, the server will configure a default Web Application. Customers upgrading to WebLogic Server 6.1 should not need to perform any registrations because this default Web Application closely approximates the document root, the `JSPServlet`, and servlet registrations performed using the `weblogic.properties` file contained in earlier versions.
- Convert your existing `weblogic.properties` file to `.xml` files using the Administration Console. See [Console Help at http://e-docs.bea.com/wls/docs61/ConsoleHelp/conversion.html](#) for more details.
- SSI is no longer supported.
- URL ACLs are deprecated. Use Servlet 2.3 features instead.
- Some information has moved from `web.xml` to `weblogic.xml`. This reorganization allows a third-party Web application based strictly on Servlet 2.2 to be deployed without modifications to its J2EE standard deployment descriptor (`web.xml`). WebLogic Server 5.1 style settings made in the `web.xml` file using `<context-param>` elements are supported for backward compatibility, but you should adopt the new way of deploying. The following sets of parameters previously defined in `web.xml` are now defined in `weblogic.xml`:

**JSP Parameters** (*keepgenerated*, *precompile compileCommand*,

*verbose, packagePrefix, pageCheckSeconds, encoding)*

**HTTP sessionParameters** (*CookieDomain, CookieComment, CookieMaxAgeSecs, CookieName, CookiePath, CookiesEnabled, InvalidationIntervalSecs, PersistentStoreDir, PersistentStorePool, PersistentStoreType, SwapIntervalSecs, IDLength, CacheSize, TimeoutSecs, JDBCConnectionTimeoutSecs, URLRewritingEnabled*)

- For more information, see [Writing Web Application Deployment Descriptors at http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html](http://e-docs.bea.com/wls/docs61/webapp/webappdeployment.html).

## XML 6.0->6.1 Parser

The XML 6.0 -> 6.1 parser has been updated and is now based on the Apache Xerces parser. The parser implements version 2 of the SAX and DOM interfaces. Users who made use of older parsers that were shipped in the `weblogic.jar` (such as Sun's Project X parser) may receive deprecation messages.

## Deprecated APIs and Features

The following APIs and features are deprecated in anticipation of future removal from the product:

- WebLogic Events

WebLogic Events are deprecated and should be replaced by JMS messages with `NO_ACKNOWLEDGE` or `MULTICAST_NO_ACKNOWLEDGE` delivery modes. See [Programming WebLogic JMS at http://e-docs.bea.com/wls/docs61/jms/fund.html#nontransacted](http://e-docs.bea.com/wls/docs61/jms/fund.html#nontransacted) for more information.

- WebLogic HTMLKona
- T3 Driver
- `-Dweblogic.management.host`

## Removed APIs and Features

The following APIs and features have been removed:

- The old administrative console GUI.  
Use the new Administration Console.
- The Deployer Tool
- WebLogic Beans
- WebLogic jHTML
- WebLogic Remote
- WorkSpaces
- WebLogic Server Tour
- T3Client
- Jview support
- SSI
- Weblogic Bean Bar
- RemoteT3
- Jview support
- Weblogic COM  
This feature relied on the Microsoft JVM (Jview), which is no longer supported.

# 4 Migrating WebLogic Server 6.0 Applications to WebLogic Server 6.1

The following sections describe steps for migrating applications from 6.0 to 6.1, as well as migration issues to keep in mind.

- Migrating an Application from 6.0 to 6.1
- JMS
- EJB 2.0
- JMX
- Apache XML Parser
- Xalan XML Parser
- Web Applications
- Security
- config\applications Directory
- WebLogic Server Clusters on Solaris
- Thread Pool Size

# Migrating an Application from 6.0 to 6.1

Use the following steps to migrate applications from WebLogic Server 6.0 to WebLogic Server 6.1:

1. Install WebLogic Server 6.1.

**Note:** Installing the new version directly over the old version is explicitly prohibited by the installer.

2. For each configuration domain that you wish to migrate to WebLogic Server 6.1, copy the domain directory to the `WL_HOME\config` directory. This directory will be used to store all of your configuration domains for WebLogic Server 6.1. If your 6.0 `config` directory is not located in the WebLogic Server 6.0 distribution, you may re-use your WebLogic 6.0 configuration in WebLogic Server 6.1.
  - a. Identify all external files referenced in `config.xml`. WebLogic Server configurations rely on a number of files that may be stored on the file system. Typically, these files are persistence repositories (logs files, file-based repositories, etc.) or utilities (Java compiler). These files can be configured using fully qualified or relative paths.

If all external files are defined using relative paths and are located in or below the domain directory, skip to Step 3.b.

For external files that are defined using relative paths that are located outside the domain directory, re-create the directory structure relative to the new `config` directory and copy the associated files into the new directories. For external files that are defined using fully qualified paths, determine whether it is appropriate to re-use these files in the WebLogic Server 6.1 deployment.

For example, logs files and persistence store can be re-used; however, you may want to update utilities such as the Java compiler to use the latest version. For files that should be updated, reconfigure the appropriate attribute using the WebLogic Server 6.0 console to use the new file or utility before proceeding to the next step.

Also consider deployment descriptors (`web.xml` and/or `weblogic.xml`) because those files may contain file paths to items such as the Java compiler.

- b. Copy the appropriate domain directory and all sub-components (directories and files) except `console.war` into the `config` directory created in Step 2.  
Make sure to move all transaction logs (\*.tlog) and any bloated message queues with the domain.
3. Edit the server start script for the new domain to use the WebLogic Server 6.1 libraries:

`CLASSPATH` -

Must include the `weblogic.jar` from the WebLogic Server 6.1 distribution

`java.security.policy` property -

Must point to the `weblogic.policy` file from the WebLogic Server 6.1 distribution

`bea.home` property -

Must point to your BEA home directory containing the `license.bea` file for WebLogic Server 6.1.

## JMS

Consider the following issues when migrating from WebLogic 6.0 to WebLogic Server 6.1.

- In WebLogic Server 6.0, the JMS documentation correctly specifies "default," "true," and "false" for the `StoreEnabled` attribute of the `JMSDestinationMBean`, but the software allowed for mixed case characters. WebLogic Server 6.1, however, requires all lowercase characters for the `StoreEnabled` settings.
- For the `Acknowledge Policy` attribute in the JMS Connection Factory, the default value of "All" is a workaround to accommodate a change in the JMS 1.0.2 specification. The `Acknowledge Policy` default setting represents a change from prior versions of JMS, which defaulted to "Previous" internally, as it did not appear as an option in the Administration Console.
  - All - acknowledge all messages ever received by a given session, regardless of which message calls the `acknowledge` method.

- Previous - acknowledge all messages received by a given session, but only up to and including the message that calls the acknowledge method. Note that although the default message driven bean (MDB) connection factory already does this, foreign connection factories may differ.

For detailed instructions on migrating previous versions of WebLogic JMS applications, see the [“Migrating WebLogic JMS Application” chapter in \*Programming WebLogic JMS\*](#).

## EJB 2.0

Sun Microsystem’s EJB 2.0 specification has changed between the 6.0 and 6.1 version of WebLogic Server, and therefore we no longer support some deployment descriptor directives. This has caused an exception to the WebLogic Server 6.0 to 6.1 migration rule that WebLogic Server 6.0 applications should work unaltered in the WebLogic Server 6.1 server. (EJB1.1 beans that worked in WebLogic Server 6.0 should work just as well in WebLogic Server 6.1 with no alteration.)

In Weblogic 6.1 a new interface, `EJBComponentMBean`, extends both `ComponentMBean` and `EJBContainerMBean`. Any custom MBeans that implement `EJBComponentMBean` (for example, `getVerboseEJBDeploymentEnabled`) have to be changed to support the additional interface when you migrate from WebLogic Server 6.0 to 6.1. When `EJBContainerMBean` is extended, `EJBComponentMBean` supplies the same methods that `EJBContainerMBean` does even though `EJBComponentMBean` is not an EJB container.

## Upgrading EJB 2.0 Session Beans and BMP Entity Beans

There have not been any major changes in the EJB 2.0 session bean or BMP entity bean model. Re-running the WebLogic EJB Compiler (`ejbc`) in WebLogic Server 6.1 ensures that your `ejb-jar.xml` file conforms to the latest EJB 2.0 DTD.

## Upgrading EJB 2.0 Message-Driven Beans (MDB)

The EJB 2.0 deployment descriptor elements in the `ejb-jar.xml` file have changed for Message-Driven beans. The `<jms-acknowledge-mode>` should be changed to `<acknowledge-mode>`, and `<jms-destination-type>` is now `<destination-type>`.

WebLogic Server 6.0 Message-Driven bean users may have a tag named `<run-as-specified-identity>` in their deployment descriptor. This tag was not supported in WebLogic Server 6.0, and its name has changed to `<run-as>` in the latest EJB 2.0 changes. Since WebLogic Server 6.0 did not support this feature, most 6.0 Message-Driven bean users probably do not need a `run-as` deployment and can remove this element from their `ejb-jar.xml` file. If you need a `run-as` identity, you must change the element to `<run-as>` in the `ejb-jar.xml` file.

## Upgrading EJB 2.0 CMP Entity Beans

The EJB 2.0 specification contains a number of changes to the CMP 2.0 model.

The first step is to ensure that your `ejb-jar.xml` file conforms to the latest EJB 2.0 DTD. There have been some changes in the element names. The `<role-source>` tag is now named `<relationship-role-source>`. The `<run-as-specified-identity>` tag is now `<run-as>`.

The latest EJB 2.0 specification introduces the concept of a local interface. EJB relationships are now based on the local interface. Any EJB that participates in a relationship must have a local interface.

Remote Relationships were removed from the EJB 2.0 specification. Relationships in WebLogic Server are now only between EJB 2.0 CMP entity beans, and the related beans must be deployed within the same EJB JAR file.

# JMX

All public WebLogic Server 6.0 MBeans and attributes are supported in WebLogic Server 6.1. However, if you are employing internal MBeans or attributes, you may encounter migration issues.

# Apache XML Parser

The XML WebLogic Server 6.0 -> 6.1 parser has been updated and is now based on the Apache Xerces 1.3.1 parser. The parser implements version 2 of the SAX and DOM interfaces. Users who used older parsers that were shipped in previous versions may receive deprecation messages.

The parsers provided in the `xmlx.jar` file have been deprecated but are included for backward compatibility. Users who prefer to use external parsers should do so through JAXP. Migration of non-JAXP code to JAXP code may involve some code adjustments.

# Xalan XML Parser

The Xalan APIs are deprecated. If you are still using those APIs and encounter difficulties, you are advised to use the JAXP API to use XSLT.

Changes were made to Apache's Xalan code to enable Xerces and Xalan to work together. You may encounter problems if you use Xalan from Apache, because it will not include these changes.

In general, it is best to use JAXP and to migrate any vendor-specific code to a neutral API such as JAXP for SAX, DOM and XSL processing.

---

# Web Applications

The

`weblogic.management.descriptors.webapp.ServletMBean.getName()` API (in WebLogic Server 6.0) has changed to `weblogic.management.descriptors.webapp.ServletMBean.getServletName()` in WebLogic Server 6.1. You will have to update your source code and recompile if you are using this interface.

With Java Servlet Specification 2.3, authorization on forward is no longer default behavior. To obtain authorization when you forward to a secure resource, add `<check-auth-on-forward>` to `weblogic.xml`.

Servlet Request and Response objects have a new API. Some serializable, lightweight implementations of these may no longer compile without implementing the new API. It is strongly recommended that you use the new Servlet 2.3 model and substitute your implementations of Servlet Request and Response objects. If you did this in WebLogic Server 6.0, you were probably relying on the undocumented, internal implementations of these objects. WebLogic Server 6.1 supports Servlet 2.3, so you should be able to take advantage of the new ServletRequest/ResponseWrapper objects.

## Security

The default condition of the server is to impose the tightest access control list (ACL) on all MBeans. Therefore, if you are making programmatic MBean changes as any non-system user you will be denied access unless you relax the restriction.

# config\applications Directory

WebLogic Server 6.1 introduces a division between runtime modes. The two modes are "development" and "production." The runtime mode is selected using a command line parameter when starting the Weblogic Admin Server (`-Dweblogic.ProductionModeEnabled=true | false`). If this parameter is not set, the server runs in development mode. In development mode the server behavior is consistent with WebLogic Server 6.0. In production mode, however, the auto-deployment feature is disabled. Deployment units in the applications directory that are not explicitly deployed in the configuration repository (`config.xml`) will not be automatically deployed. Note that in WebLogic Server 6.1 the default (mydomain) and Pet Store configurations are shipped in production mode. The examples configuration is shipped in development mode. This feature only works on the Admin Server, because the AppManager thread that polls the applications directory for changes is only created on Admin servers.

## WebLogic Server Clusters on Solaris

Certain applications (heavy EJB apps) deployed in a WebLogic Server cluster on Solaris will perform better using the client JVM rather than the server JVM. This is especially true under heavy loads.

## Thread Pool Size

In WebLogic Server 6.0, the number of worker threads was specified via the `ThreadPoolSize` parameter on the server MBean. Starting in WebLogic Server 6.1, the number of worker threads is defined via an `ExecuteQueue` on the Server MBean.

WebLogic Server 6.1 provides a migration path for this parameter, so that if it is specified in the `config.xml`, or if it is passed to the client or server on the command line (`-Dweblogic.ThreadPoolSize=<xx>`), WebLogic Server migrates your `ThreadPoolSize` to the `ThreadCount` setting automatically.

In WebLogic Server 6.1 Service Pack 1, there was a bug in the console that displayed an incorrect value (the default value) for the `ThreadCount`. However, this was only a bug in the console, and the correct number of threads was being used by the client/server.

Setting the number of worker threads :

In WebLogic Server 6:0

```
<Server
Name="myserver"
ThreadPoolSize="23"
...
/Server>
```

Starting in WebLogic Server 6.1:

```
<Server
Name="myserver"
... >
<ExecuteQueue
Name="default"
ThreadCount="23" />
/Server>
```

To change the thread count value via the console:

1. In the console, select **Servers > myServer > Monitoring >** .
2. Click on **Monitor all Active Queues**.
3. Click on **"default"** queue (a list of threads and what they are doing appears).
4. Click on **Configure Execute Queues** (at the top of the page).
5. Click on **"default"** queue.
6. Enter the number of threads associated with this server.
7. Restart the server to make changes take effect.

## **4** *Migrating WebLogic Server 6.0 Applications to WebLogic Server 6.1*

---

# 5 Notes and Problems

The following sections describe known problems with this software release.

- “Connector Known Problems” on page 5-3
- “Console Known Problems” on page 5-3
- “Core Known Problems” on page 5-4
- “EJB Known Problems” on page 5-6
- “Examples and Pet Store Demo Known Problems” on page 5-10
- “Installer Known Problems” on page 5-15
- “JCA Known Problems” on page 5-17
- “JDBC and jDrivers Known Problems” on page 5-17
- “JMS Known Problems” on page 5-20
- “JSP Known Problems” on page 5-20
- “JVM Known Problems” on page 5-22
- “RMI Known Problems” on page 5-25
- “RMI-IIOP Known Problems” on page 5-25
- “Security Known Problems” on page 5-26
- “Server Known Problems” on page 5-28
- “Servlet and Web Application Known Problems” on page 5-31
- “System Administration Known Problems” on page 5-34
- “Tools Known Problems” on page 5-38

- “Web Services Known Problems” on page 5-38
- “WebLogic Tuxedo Connector Known Problems” on page 5-41
- “XML Known Problems” on page 5-43
- “ZAC Known Problems” on page 5-44

---

# Connector Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
098342	The WebLogic Server 6.1 JCA implementation is not compatible with the IBM CICS resource adapter. BEA's implementation of the <code>javax.resource.spi.ConnectionManager</code> interface is not directly returning the object returned by the <code>getConnection()</code> method of the resource adapter's implementation of the <code>javax.resource.spi.ManagedConnection</code> interface. It returns an instance of a class dynamically generated by the <code>java.lang.reflect.Proxy</code> class instead.

---

# Console Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
053277	If you use Netscape to bring up the Administration Console and try to edit a template for an LDAP Realm configuration, using the arrows or the bottom scroll bar to go left or right in the Configuration Data window causes the displayed data to look garbled. If you click the Apply button, no corrupt data is saved in <code>config.xml</code> , but it is difficult to edit the template.

---

---

## Core Known Problems

---

Change Request Number	Description
CR093104	<p>In SP04 and SP05, the NT Performance pack causes thread deadlock under some circumstances. The problem was exhibited under light loads during stress tests of a .jsp's accessing a database. Thread dumps showed the majority of "default" execute threads stuck in "waiting for monitor entry":</p> <pre data-bbox="400 540 1257 1179"> "ExecuteThread: '10' for queue: 'default'" daemon prio=5 tid=0x273fb548 nid=0xad0c waiting for monitor entry [0x2810f000..0x2810fdc4]     at     weblogic.socket.NTSocketMuxer.initiateIO(NTSocketMuxer.java:483)     at     weblogic.socket.NTSocketMuxer.read(NTSocketMuxer.java:474)     at     weblogic.servlet.internal.MuxableSocketHTTP.requeue(MuxableSocketHTTP.java:244)     at     weblogic.servlet.internal.ServletResponseImpl.send(ServletResponseImpl.java:1094)     at     weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2364)     at     weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)     at     weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) </pre> <p>Customers experience such deadlocks should create a <code>.hotspot_compiler</code> file in the directory where you invoke the JVM and put the following line in that file :</p> <pre data-bbox="400 1256 1130 1279">exclude weblogic/socket/NTSocketMuxer processSockets</pre>
CR106616	<p>There is a problem in Netscape Version 4.79 running on Linux AS2.1 and HP-UX. After several operations on the WebLogic Server Administration Console, the browser shuts down with a bus error, sometimes with a core dump.</p>

---

---

Change Request Number	Description
-----------------------	-------------

---

CR179419

In WebLogic Server 6.1, when a Managed Server is configured to use the pure-Java socket reader implementation, rather than native socket readers, execute threads are allocated to act as socket reader threads. You configure the pure-Java socket reader implementation by setting the the server instance's `NativeIOEnabled` attribute to `false`, and the number of threads that act as socket reader threads is a function of the server instance's `ThreadPoolPercentSocketReaders` value and the execute queue's `ThreadCount` attribute.

Under these circumstances, the value of `ThreadCount` is not honored. The server instance issues the following warning before the configured thread count is reached:

```
<Apr 22, 2004 8:42:54 PM JST> <Warning> <JavaSocketMuxer> <There are: '3' active sockets, but the maximum number of socket readers allowed by the configuration is: '2', you may want to alter your configuration.>
```

This problem results from an error in thread pool initialization logic. The problem does not occur for Administration Servers.

The work-around for this problem is to assign the thread pool size using this command line option when starting the Managed Server:

```
-Dweblogic.ThreadPoolSize=xxx
```

where `xxx` is the desired thread count.

For example:

```
java ... -Dweblogic.ThreadPoolSize=15 ... weblogic.Server
```

---

## EJB Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
035884	The maximum number of Enterprise JavaBeans (EJBs) that can be deployed on one WebLogic Server instance is 390. This is due to a Sun bug and is a limitation caused by the 1.3 JDK; it may be fixed by the 1.3.1 release of the JDK. The Sun Bug ID is: 4390238.
051543	If you try to deploy a completely incorrect EJB, the EJB container throws an exception from its deploy method. However, if you click on that EJB in the console, it still has the Deployed box checked.

---

---

**Change Request    Description**  
**Number**


---

061938

In certain conditions, EJB QL queries can return spurious duplicates that are the results of SQL cross products. This can occur under the following conditions:

- The EJB-QL query contains path expressions that navigate relationships; this generates multiple tables in the generated SQL SELECT clause.
- The WHERE clause contains OR operands which navigate relationships, and not all of the path expression in the OR operands map to all of the tables in the generated SQL SELECT clause for the query; this may cause a cross product to show up in the results for that OR operand.

The following example illustrates the problem:

EJB QL:

```
SELECT OBJECT(c)
FROM CustomerBean AS c, IN(c.accounts)accts
WHERE c.name = '100' OR c.accts.bal = 300
```

DATA: customer '100' exists but has no accounts

EXPECTED RESULT: customer '100' from clause #1

ACTUAL RESULT: customer '100' X number of accts

SQL:

```
DROP TABLE thorick_customers;
```

```
CREATE TABLE thorick_customers (
  cust_name VARCHAR(10),
  cust_interests VARCHAR(10),
  cust_rating INTEGER, acct_id INTEGER,
  PRIMARY KEY (cust_name));
```

```
DROP TABLE thorick_accounts;
```

```
CREATE TABLE thorick_accounts
(acct_id INTEGER,
 bal FLOAT,
 PRIMARY KEY (acct_id));
```

```
INSERT INTO thorick_accounts VALUES (100, 100.0);
INSERT INTO thorick_accounts VALUES (200, 200.0);
INSERT INTO thorick_accounts VALUES (300, 300.0);
INSERT INTO thorick_accounts VALUES (400, 400.0);
INSERT INTO thorick_accounts VALUES (500, 500.0);
INSERT INTO thorick_customers VALUES('100', 'jazz', 2, null);
INSERT INTO thorick_customers VALUES('900', 'punk', 3, 400);
```

```
SELECT WL0.cust_name, WL0.cust_interests, WL0.acct_id
FROM thorick_Customers WL0, thorick_Accounts WL1
WHERE WL0.cust_name = '100'
OR ( w11.bal = 300 AND w10.acct_id=w11.acct_id );
```

CUST_NAME	CUST_INTER	ACCT_ID
-----	-----	-----
100	jazz	

<b>Change Request Number</b>	<b>Description</b>
062927	<p>Requesting an EJB served by WLE 5.1 from a JSP served by WLS 6.1 may give the following exception:</p> <pre>java.lang.NoClassDefFoundError</pre> <p>Add the location of the WLE 5.1 <code>wlej2eecl.jar</code> file to the CLASSPATH of your WLS 6.1 <code>startWeblogic.cmd</code> or <code>startWeblogic.sh</code> script.</p>
079539	<p>WebLogic Server 6.1 sp3 is not compatible with Application Integration in WLI 2.1 sp1. Please contact <a href="#">BEA Customer Support</a> for more information.</p>
088461	<p><code>ejbc</code> behavior varies between WebLogic Server 6.1 SP02 and SP03. In SP02, running <code>ejbc</code> on a bean before deploying it to the server worked as expected, while in SP03, the same action on the bean sometimes fails with an <code>OutOfMemory</code> exception if hot deployed without precompiling with <code>ejbc</code>.</p> <p>The workaround is to use:</p> <pre>java -verbosegc weblogic.ejbc -compiler javac -J-mx128m entityBeans.jar final.jar</pre>
089847	<p>Many of the WebLogic Server EJB deployment properties have default values that are optimized for performance. In some cases, these default values are not compliant with the EJB specification. Should you wish to make WebLogic Server compliant with the EJB specification, you must set the following properties:</p> <p>In <code>weblogic-ejb-jar.xml</code> set "enable call by reference" to False.</p> <p>In <code>weblogic-cmp-jar.xml</code>, set "include-updates" to True</p> <p>In <code>weblogic-cmp-jar.xml</code>, set "check exists on method" to True.</p>

---

Change Request Number	Description
097323	<p>There is a known issue with deploying or redeploying EJBs to a single server instance in a cluster—referred to as pinned deployment—if the .jar file contains contain uncompiled classes and interfaces.</p> <p>During deployment, the uncompiled EJB is copied to each server instance in the cluster, but it is compiled only on the server instance to which it has been deployed. As a result, the server instances in the cluster to which the EJB was not targeted lack the classes generated during compilation that are necessary to invoke the EJB. When a client on another server instance tries to invoke the pinned EJB, it fails, and an <code>Assertion</code> error is thrown in the RMI layer.</p> <p><b>Workaround:</b> If you are deploying or redeploying an EJB to a single server instance in a cluster, compile the EJB with <code>appc</code> or <code>ejbc</code> before deploying it, , to ensure that the generated classes copied to all server instances available to all nodes in the cluster.</p>
111022	<p>When using a non-transaction-enabled <code>DataSource</code>, a "java.sql.SQLException: ORA-01002" error will occur when creating a <code>CMP</code> entity bean which has a field mapped to Oracle <code>CLOB/BLOB</code>. The EJB container inserts a row which has <code>CLOB/BLOB</code> columns as follows:</p> <ol style="list-style-type: none"><li>1. insert into the table initializing the LOB using <code>EMPTY_BLOB()</code> or <code>EMPTY_CLOB()</code>.</li><li>2. retrieve the LOB locator using <code>select for update</code> statement.</li><li>3. modify the LOB.</li></ol> <p>Although the connection from a non-transaction-enabled <code>DataSource</code> is in auto-commit mode, Oracle LOB locators that are used to write cannot span the transactions. So the transaction is closed at step 1 and then <code>ORA-1002</code> occurs at step 2.</p> <p>Workaround:</p> <p>You need to use a transaction <code>DataSource(TxDataSource)</code> when using Oracle <code>CLOB/BLOB</code> with <code>CMP</code>.</p>

---

# Examples and Pet Store Demo Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
043156	<p>The <code>ejbmanagedclient.jsp</code> relies on a variable that is set to <code>t3://localhost:7001</code> and does not get the port number dynamically. Thus, when the page is requested, it cannot find the server. The exception is thrown, although the stack trace is not printed out. The page shows up as:</p> <pre>-----UnexpectedError-----</pre> <p>Customers running this example on a different port than 7001, will need to edit the <code>jsp</code> file.</p> <p>The variable you must edit is in this section:</p> <pre>&lt;!-- Here, we declare a class method --&gt; &lt;%! String pagetitle = "JSP example using EJB-managed persistence"; String url = "t3://localhost:7001";</pre> <p>Edit the URL to point to the correct server/port number that has been installed.</p>
044641	<p>There is some missing information in <code>examples/ejb/sequence/userDesignated/package-summary.html</code>. Make sure that you use the <code>table.ddl</code> file, included in the example's package, as the source for the table that needs to be built for this example. Also, make sure that table is named <code>"NAMED_SEQUENCE_TABLE"</code> not <code>"NAMED_SEQUENCE-TABLE"</code>.</p>
047654	<p>In the <code>dbKona</code> examples, a pool connection cannot be made and an error is thrown when trying to boot the example server.</p>
047928	<p>When running the Web Application security example, there is no link in the console for creating a group and adding members to it. Also, on the group window, clicking on Name or Members causes a 500 Internal Server Error.</p>

---

Change Request Number	Description
048358	The <code>security/rdbmsrealm/rdbmsrealm.ddl</code> and <code>utils/ddl/demo.ddl</code> examples have problems with improper comment symbols.
052556	The WebLogic Tuxedo Connector CORBA <code>simpappcns</code> example fails when run in the examples domain after running the WLEC examples. The problem is a conflict with the <code>Simpapp</code> client stubs left in the <code>SERVER_CLASSES</code> directory from the WLEC examples. Before running the <code>simpappcns</code> example, check the <code>SERVER_CLASSES</code> environment variable. If the WLEC examples have been run, delete all <code>simpapp</code> client stubs from <code>SERVER_CLASSES</code> .
053356	The example <code>samples/examples/wtc/corba/simpappcns</code> does not have a <code>run.sh</code> UNIX script and will not work on UNIX machines. This problem was solved in WebLogic Server 6.1 SP04. See “053356”.
062545	WebLogic Server domain running Solaris <code>simpappcns</code> can't call Tuxedo Corba Server domain running on NT. This problem was solved in WebLogic Server 6.1 SP03. See “062545”.
062799	The documentation for three of the WIRELESS examples contains an error in Step 7 of the instructions. The examples in question are <code>helloWorld</code> , <code>stockDemo</code> and <code>travelDemo</code> . Step 7 should read: <code>Path=./config/examples/helloWorld</code> rather than: <code>Path=./config/examples</code> If uncorrected, the examples will not find the deployed applications.
064001	<code>examples.webservices.message.package-summary.html</code> contains an incorrect instruction. Step 13 says “Right-click the Destinations node and choose Configure a new <JMSTopic>” from the drop-down list.” JMSTopic should read JMSQueue.
078441	Some examples fail when a data source is re-targeted. If, after re-targeting a data source, an example fails with the error message, “Cannot obtain connection after X seconds”, make sure to restart the examples server in order for the change to take effect.

Change Request Number	Description
CR102138	<p>In WebLogic Server 6.1 SP05, the JSSE version of the SSLClient sample code example may not work because the <code>jssecacerts</code> file does not contain the correct <code>ca.pem</code> file.</p> <p>The work-around for this problem is to add the <code>ca.pem</code> file provided in the WebLogic Server distribution to the <code>jssecacerts</code> file.</p> <p>To add <code>ca.pem</code> to the <code>jssecacerts</code> file:</p> <ol style="list-style-type: none"><li>1. Open a command window and change directory to: <pre>WL_HOME\samples\examples\security\sslclient&gt;</pre></li><li>2. At the prompt, enter: <pre>keytool -v -keystore jssecacerts -trustcacerts -import -file %WL_HOME%\config\examples\ca.pem</pre></li><li>3. When prompted for a password, enter: <pre>changeit</pre></li></ol>
CR103521	<p>The documentation for the <code>dbkona</code> examples omit the following additional steps, which are necessary to compile and run the examples:</p> <ol style="list-style-type: none"><li>1. Select the appropriate <code>weblogicoci37.dll</code> based on the version of Oracle you are using. Add the path to the correct DLL to your <code>PATH</code> environment variable.</li><li>2. Run the following command to load the database: <pre>java utils.Schema "jdbc:weblogic:oracle" weblogic.jdbc.oci.Driver -s oracle_server -u scott -p tiger -verbose utils\ddl\demo.ddl</pre></li><li>3. Before compiling the code for an example, edit all of the example's Java files to change the <code>server</code> property to the value of <code>oracle_server</code> that you specified in the previous step.</li></ol>

---

Change Request Number	Description
-----------------------	-------------

---

CR103547	<p>dbkona examples fail to start on HP and Solaris. When a dbkona example is launched, this error message results:</p> <pre>Starting Loading jDriver/Oracle . . . . . Trouble while executing example java.sql.SQLException: System.loadLibrary(weblogicoci37) threw java.lang.UnsatisfiedLinkError: /space/bt/ weblogic/load4/bea/wlserver6.1/lib/hpux11/oci817_8 /libweblogicoci37.sl: specified library, or one of its dependencies, does not exist. at weblogic.jdbc.oci.Driver.loadLibraryIfNeeded (Driver.java:243 at weblogic.jdbc.oci. Driver. connect (Driver.java:79)at examples.dbkona.query. main(query.java:118)</pre> <p>The problem occurred because jDriver is looking for libclntsh.so.8.0 file, which should be located in the ORACLE_HOME/lib directory. jDriver did not find the file because ORACLE_HOME was not defined. The example instructions do not include instructions to:</p> <ul style="list-style-type: none"><li>■ ensure that the ORACLE_HOME environment variable is defined</li><li>■ ensure that the ORACLE_HOME/lib directory contains the libclntsh.so.8.0 file</li><li>■ ensure that the ORACLE_HOME/lib directory is in the LD_LIBRARY_PATH.</li></ul> <p>These steps are required to run the dbkona examples</p>
----------	--

---

---

Change Request Number	Description
-----------------------	-------------

---

CR127253	<p>The <code>examples.security.acl</code> example does not trap the error that results when the user does provide the appropriate command line options. Instead, a stack trace is displayed.</p> <p>The package summary for the <code>examples.security.acl</code> example says:</p> <p>"user and password are the username and password for joeuser. If the <code>-user</code> and <code>-pass</code> command line options are not specified, the JNDI <code>Initial_Context</code> defaults to the user "guest" with the password "guest". In this case, the <code>AltClient</code> client will fail because the <code>aclexample</code> ACL does not assign permissions to the user "guest."</p> <p>Running the example without following these instructions resulted in this error stack trace:</p> <pre>C:\610sp6\wlserver6.1\samples\examples\security\ac l&gt;java -Dweblogic.security.SSL .ignoreHostnameVerification=true examples.security.acl.AltClient t3s://localhost :7002 -sslCert demokey.pem;democert.pem -servername weblogic.bea.com javax.naming.AuthenticationException. Root exception is java.lang.SecurityException: Authentication for user null denied in realm weblogic Start server side stack trace: java.lang.SecurityException: Authentication for user null denied in realm weblog ic at weblogic.security.acl.Realm.authenticate(Realm.jav a:212) at weblogic.security.acl.Realm.getAuthenticatedName(R ealm.java:233) at weblogic.security.acl.internal.Security.authentica te(Security.java:17 1) at weblogic.common.internal.RMIBootServiceImpl.authen ticate(RMIBootServi ceImpl.java:47) at weblogic.common.internal.RMIBootServiceImpl_WLSkel .invoke(Unknown Sou rce) at weblogic.rmi.internal.BasicServerRef.invoke(BasicS erverRef.java:360) at....</pre>
----------	---

---

# Installer Known Problems

Change Request Number	Description
039773	<p>The following misleading error message occurs when there is not enough space in /temp to run the installer:</p> <pre>Preparing to install...</pre> <p>The included VM could not be extracted. Please try to download the installer again and make sure that you download using 'binary' mode. Please do not attempt to install this currently downloaded copy.</p> <p>By default the installer tries to unpack all the installer files in the /temp directory. If there isn't enough temporary space there, you cannot install and you get this message. Make room in the /temp directory for the installer files.</p>
040353	<p>Installation problems can occur on HP-UX and Solaris machines because shared libraries do not have execute permission. File permission changes made by the WebLogic Server installer are reverted by installer imposed standard attributes. A workaround is to issue the <code>chmod +x</code> command on all the shared library objects (under <code>lib/hpux11</code> and <code>lib/solaris</code>).</p>
043357	<p>If you are getting the following error on Solaris 2.6 or 2.7 UNIX systems:</p> <pre>UnsatisfiedLinkError in AWT applications (libmawt.so: open failed: No such file for directory). The policytool will not open. The Java Plug-in Control Panel will not open.</pre> <p>Usually you might workaround this problem by modifying the <code>LD_LIBRARY_PATH</code>. For example:</p> <pre>setenv LD_LIBRARY_PATH /usr/j2se/jre/lib/i386/motif12:\$LD_LIBRARY_PATH</pre> <p>But, in some cases with problems with the <code>libmawt.so</code> files, you may require a patch from Sun: see <a href="http://java.sun.com/j2se/1.3/install-solaris-patches.html">http://java.sun.com/j2se/1.3/install-solaris-patches.html</a> for Sun's patch information.</p>

<b>Change Request Number</b>	<b>Description</b>
053469 & 055598	When using the Administration Console, add the line, <code>acl.modify.weblogic.admin=Administrators</code> , to the <code>filerealm.properties</code> to grant permissions to users and/or groups. This ACL was in WebLogic Server 6.0, 6.0 Service Pack 1 and 6.0 Service Pack 2.
054058	When encountering an unsupported browser, <code>beaexec.exe</code> may result in WebLogic Server crashing.
087287	When installing on Solaris, if <code>\$DISPLAY</code> is not set, the installer throws the following exception:  Invocation of this Java Application has caused an InvocationTargetException.  This application will now exit.  To work around this problem, set <code>\$DISPLAY</code> the IP address of the host machine.
091420	WebLogic Server 6.1 SP04 failed to start if the "Enable Post-Bind UID" option for a Unix Machine was checked. (This attribute can be configured in the Machines node of the Administration Console; its purpose is to return the Unix UID a server instance running a UNIX machine will run under after it has carried out all privileged startup actions).  This problem is resolved in WLS 6.1 SP05.  See "CR091420" on page 6-155.
092326	When installing WebLogic Server SP04 using the upgrade installer from WebLogic Server 6.1 SP03, an earlier version of <code>beasvc.exe</code> is installed than if you use the full installer for WebLogic Server SP04.  To work around this problem, use the full installer for WebLogic Server SP04.
102815	There is a problem with the Install Anywhere installer that requires WebLogic Server 6.1 to be installed using Windows 2000 compatibility mode. For information, see <a href="#">Installing on Windows XP</a> .

## JCA Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
049344	The current BEA WebLogic J2EE Connector architecture implementation does not provide the ability to detect the errors discussed in section 6.8 (Scenarios: Local Transaction Management) of the J2EE Connector Specification, Version 1.0, Proposed Final Draft 2.

---

## JDBC and jDrivers Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
033423	dbKona problems exist with <code>QueryDataSet</code> and the <code>OCI805_8</code> driver. The problems are fixed on Windows NT, but this is still a problem on HP-UX and Solaris.
035301	There are problems using the RMI driver with Sybase connection pools.
035530	Problems with enabling OS authentication and using Oracle 8.1.6/oci8 with the <code>jDriver</code> are fixed on NT and HP-UX, but are still an issue on Solaris. An error message that reports this problem: OS level authentication is not currently supported due to a defect in OCI 8 libraries.

---

<b>Change Request Number</b>	<b>Description</b>
040413	<p>Cluster hanging problems have been observed in some instances on Solaris machines. This appears to be caused by a <code>libthread</code> deadlock between <code>SIGALRM</code> and <code>SIGLWP</code>. The deadlock manifests itself when the server is attempting a JDBC connection. To help avoid the problem, use the Administration Console to set the <code>InitialCapacity</code> attribute equal to the <code>MaxCapacity</code> attribute in the <code>JDBCConnectionPool</code> entry.</p> <p>A similar problem was previously logged as a Sun Microsystems Bug (ID 4141709 <code>libthread</code> deadlock between <code>SIGALRM</code> and <code>SIGLWP</code>), against Solaris 2.6. There is currently no mention of the problem for Solaris 2.7. For Solaris 2.6, the patch for this problem is 105569-16.</p>
043224	<p>Oracle supports of its own class of <code>Array</code> (<code>oracle.sql.Array</code>) through its <code>ThinDriver</code>. If you attempt to use these classes in an application that gets its connection using <code>WebLogic Server JDBC</code>, they will not work.</p>
048219	<p>Flush doesn't seem to "commit" with <code>RMI Clob</code> character write. This only seems to affect flushing when you are using these write methods:</p> <ul style="list-style-type: none"><li>■ <code>write(char[] cbuf)</code></li><li>■ <code>write(char[] cbuf, int off, int len)</code></li><li>■ <code>write(String str)</code></li><li>■ <code>write(String str, int off, int len)</code></li></ul> <p>It works fine with:</p> <ul style="list-style-type: none"><li>■ <code>write(int c)</code></li></ul> <p>No problems with flushing occur when using an <code>Ascii Stream</code> with a <code>Clob</code>. Also, to commit your write, you have to explicitly close the stream. This may only be a problem if you wanted to read back from the <code>Clob</code> before you were ready to close out your write stream.</p>
049519	<p>Due to a bug in Oracle's thin driver, the following method is not compliant with the <code>java.sql.PreparedStatement</code> API:</p> <pre>PreparedStatement.setTimeStamp(int param index, Timestamp x, Calendar cal)</pre> <p>The timestamp value is not adjusted according to the <code>TimeZone</code> specified in the <code>Calendar</code> object. The default <code>TimeZone</code> is always used. Oracle may fix this bug in future releases of the thin driver.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

127348	<p>When using the Oracle 8.1.7 Thin driver included with WebLogic Server 6.1SP6, the following exception may appear intermittently:</p> <pre>java.sql.SQLException: Io exception: Protocol violation</pre> <p>This may be caused by an error in the driver. The exception does not occur with a later version of the driver available from Oracle. WebLogic Server 6.1SP6 includes the 8.1.7.0.0 version of the driver. Version 8.1.7.1.0 is currently available from Oracle.</p> <p>Workaround: Get the new version of the driver from Oracle and add it (classes12.zip) to your classpath in front of weblogic.jar.</p>
--------	---

---

## JMS Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
037683	The JMS Topic configuration tab in the console accepts any alphabetic characters as valid values for a Multicast Address. It should validate the value that is entered against an Internet Address format.
049229	You may experience problems with JMS if you run the server with the unsupported JDK 1.3.0. This problem occurs when restarting the server when there are existing messages in the JMS store -- these messages are ignored. It is recommended that you use the 1.3.1 JDK that is bundled with the server.
083538	In a multi-server configuration, an application or EJB on one server cannot lookup and get a connection from a <code>TxDatasource</code> on another server. The application or EJB must use a <code>TxDatasource</code> on the same server. This is a system limitation.

---

## JSP Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
049854	The <code>ejb2</code> JSP tool does not support EJB 2.0 local interfaces.
075671	When using Java classes such as <code>FileInputStream</code> or <code>ArrayList</code> in JSPs, you will not receive a <code>NoClassDefFoundError</code> , even if you do not import these classes into the JSP.

---

---

Change Request Number	Description
104828	<p>Struts 1.02 recommends moving all JSPs below the WEB-INF directory of the Web application to prevent direct access. This ensures that JSPs can only be forwarded to from Action classes. Any file located inside the WEB-INF folder cannot be accessed directly.</p> <p>Configuration: WebLogic Server 6.1 SP2 or WebLogic Server 6.1 SP4.</p> <p>Solution S-12864 suggests a workaround using servlet forward.</p>
CR127701	<p>There is a known security vulnerability in JDK 1.3.1 and 1.4.1. This vulnerability could cause a JVM to crash when attempting to parse malformed XML or XSLT, resulting in a Denial of Service on the web or application server. BEA recommends installing the patch provided by Sun Microsystems to resolve this known security vulnerability.</p> <p>With the patch, more rigorous JSP parsing is performed. As a result, code that compiled under JDK 1.3.1 and 1.4.1 may not compile after installation of the patch. For information about how JSPs in the Petstore sample application were upgraded to avoid compilation errors, see <a href="#">WebLogic Server 8.1 Upgrade Guide</a> at <a href="http://e-docs.bea.com/wls/docs81/upgrade/upgrade6xto81.html#1062978">http://e-docs.bea.com/wls/docs81/upgrade/upgrade6xto81.html#1062978</a>.</p>

---

# JVM Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
----------------------------------	--------------------

---

CR099314

There is a problem with JDK 1.3.1\_04 and the `nohup` command, as described at

<http://developer.java.sun.com/developer/bugParade/bugs/4755829.html>.

The `nohup` command starts failing; the WebLogic Server process exits when the parent shell exits, regardless of the invocation of `nohup`. The problem was confirmed for `ksh` (the default Solaris 8 shell), and may exist for other shells. The problem does not occur with the `bash` shell.

Workaround:

- Use “`(startWeblogic >output 2>&1 &)`” instead of “`nohup startWeblogic &`”.
-

Change Request Number	Description
CR100564	<p>Due to an interoperability problem between WebLogic Server 6.1 SP04 and WebLogic Server 8.1 calls from an RMI client on WebLogic Server 6.1 SP04 fails to get the correct exception thrown by an RMI object on WebLogic Server 8.1. This error resulted on WebLogic Server 6.1 SP04:</p> <pre> &lt;Mar 8, 2003 12:36:35 PM PST&gt; &lt;Error&gt; &lt;NT Performance Pack&gt; &lt;failure in processSockets() - GetData: 'weblogic.socket.NTSocketMuxer\$GetData - native pointer: '272329000', numBytes: '0'' java.lang.ClassCastException: weblogic.iiop. LocateRequestMessage at weblogic.iiop.EndPoint Impl.cleanupPendingResponses(EndPointImpl.java:6 59) at weblogic.iiop.ConnectionManager.got ExceptionReceiving(ConnectionManager.java:273) at weblogic.iiop.MuxableSocketIIOP.endOfStream (MuxableSocketIIOP.java:681) at weblogic.socket. NTSocketMuxer.processSockets(NTSocketMuxer.java: 657) at weblogic.socket.SocketReaderRequest. execute(SocketReaderRequest.java:24) at weblogic. kernel.ExecuteThread.execute(ExecuteThread.java: 139) at weblogic.kernel.ExecuteThread.run (ExecuteThread.java:120) &gt; &lt;Mar 8, 2003 12:41:35 PM PST&gt; &lt;Error&gt; &lt;NT Performance Pack&gt; &lt;failure in processSockets() - GetData: 'weblogic.socket. NTSocketMuxer\$GetData - native pointer: '272089304', numBytes: '0'' java.lang.ClassCastException: weblogic.iiop. LocateRequestMessage at weblogic. iiop.EndPointImpl.cleanupPendingResponses(EndPoi ntImpl.java:659) at weblogic.iiop.Connection Manager.gotExceptionReceiving(ConnectionManager. java:273) at weblogic.iiop.MuxableSocketIIOP. endOfStream(MuxableSocketIIOP.java:681) at weblogic.socket.NTSocketMuxer.processSockets(NTS ocketMuxer.java:657) at weblogic.socket.Socket ReaderRequest.execute(SocketReaderRequest.java:2 4) at weblogic.kernel.ExecuteThread.execute (ExecuteThread.java:139) at weblogic.kernel. ExecuteThread.run(ExecuteThread.java:120) </pre>

## Plug-Ins Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
CR172808	<p>WebLogic Server 6.1 and Apache plug-in are certified on Solaris 2.6, but the WebLogic Server 6.1 SP5 plug-in fails to start with the following linkage error:</p> <pre># ./apachectl startssl</pre> <p>Syntax error on line 216 of /usr/local/apache3/conf/httpd.conf: Cannot load /usr/local/apache3/libexec/mod_wl_ssl.so into server: ld.so.1: /usr/local/apache3/bin/httpd: fatal: libpthread.so.1: version `SUNW_1.2' not found (required by file /usr/local/apache3/libexec/mod_wl_ssl.so) ./apachectl startssl: httpd could not be started</p> <p>As of WebLogic Server 6.1 SP5, plug-in binaries require libthread.so.1 (SUNW_1.2), but Solaris 2.6 has an older version of it (SUNW_1.1, SUNW_0.9).</p> <p><b>WORKAROUND:</b> Statically link libweblogic.a, instead of mod_wl.so/mod_wl_ssl.so. This is a system limitation.</p>

---

---

# RMI Known Problems

Change Request Number	Description
CR100713	<p>WebLogic Server 6.1 SP04 and earlier does not interoperate with WebLogic Server 8.1. Transactions between 6.1 and 8.1 server instances timeout with an <code>ArrayIndexOutOfBoundsException</code> on the 6.1 server instance.</p> <p>There is an RJVM/RMI problem when a 6.1 server instance performs a lookup of the 8.1 server instance's coordinator remote object. When the 6.1 server attempts an <code>ack</code>, the operation fails, the 8.1 coordinator fails to resolve the transaction and throws a <code>commit timeout</code> exception to the client.</p> <p>For example: a client and remote object startup class are deployed on 8.1 and 6.1 servers. The client looks up the remote object on the 8.1 server and invokes a remote method specifying the URL of the 6.1 server. The 8.1 method looks up the remote object on the 6.1 server and invokes a method with the URL of the 8.1 server. The 6.1 remote method looks up the remote object on the 8.1 server and invokes a method. The problem occurs when bootstrapping and sending the JVMID. There is no way of telling who the remote peer is.</p> <p>This problem is resolved in WebLogic Server 6.1 SP05.</p>

---

# RMI-IIOP Known Problems

Change Request Number	Description
CR126991	<p>There is an interoperability problem between WebLogic Server 6.1 SP05 and WebLogic Server 6.1 SP06 that may result in a <code>ClassCastException</code>. The error is a result of casting <code>IIOPRemoteRef</code> to a <code>IOR</code> during <code>replaceObject</code>. A patch for WebLogic Server 6.1 SP05 is available.</p>

---

# Security Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
048159	When a Log Audit Provider is being used, no messages are being sent to the WebLogic Log File notifying you that auditing is turned on.
051215	<p>When entering the system password during startup for WebLogic Server, if you use a CTRL/C sequence to interrupt the WebLogic Server startup, the terminal window stays in no-echo mode and characters that are typed are not echoed.</p> <p>If possible, wait to interrupt the WebLogic Server startup until after you enter the system password. If you do use a CTRL/C sequence when entering the system password, reset the terminal window or close the terminal window and start another window.</p>
051315	<p>The Node Manager in WebLogic Server cannot process the private keys created by the Certificate Request Generator servlet. The <code>wlkeytool</code> converter tool has been provided to solve this problem.</p> <p>Use the <code>wlkeytool</code> converter tools as follows:</p> <pre>wlkeytool inputkey.pem outputkey.pem</pre> <p>where</p> <p><i>inputkey.pem</i> is the name of the private key created by the Certificate Request Generator servlet.</p> <p><i>outputkey.pem</i> is the name of the converted private key.</p> <p>If the private key is password protected (that is, if you specified a password in the Private Key Password field in the Certificate Request Generator servlet), the <code>wlkeytool</code> converter tool prompts you for the password. If the private key is unprotected, press Enter.</p> <p>The <code>wlkeytool</code> converter tool prompts you for a password for the converted private key. A password must be supplied, or the <code>wlkeytool</code> converter tool will not create an unprotected private key.</p> <p>A deployed WebLogic Server does not support the converted private key created by the <code>wlkeytool</code> converter tool, so you should retain the private key created by the Certificate Request Generator servlet and use it with your deployed WebLogic Servers.</p>

---

---

Change Request Number	Description
053390	<p>If you are using an RDBMSRealm with a Caching Realm enabled, and the DatabasePassword is not specified in the RDBMSRealm tag in the config.xml file, the server halts with the error "&lt;wlsstartuperror1&gt;" and does not startup.</p> <p>Workaround: Open the config.xml file and add the DatabasePassword in clear text to the RDBMSRealm tag.</p>
053409	<p>To use the Java Security Manager with WebLogic Server, specify the -Djava.security.manager property when starting WebLogic Server.</p> <p>The included startup scripts specify a policy file but no security manager. Therefore the policy file ends up getting ignored. You must set a security manager for the policy file to become effective.</p>
078619	<p>Group Membership caching is not using the value set in GroupMembershipCacheTTL. It is instead using the value set in GroupCacheTTLPossible.</p>
078894	<p>When FlatGroup.clearCache() is invoked to refresh the cache, and isMember is invoked for a second time, isMember is returning false. It is still using old cache data.</p>
087225	<p>In WebLogic Server 6.1 SP04 and later, the LDAP realm V2 has issues with the Open LDAP Server when running the getGroups() method for large numbers of groups (more than 300). This problem is due to caching bugs in Open LDAP.</p> <p>Workaround: Do not try to obtain more than 300 groups when using the getGroups() method to retrieve groups from the Open LDAP server.</p>

---

---

# Server Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
039575	If your servlet makes an HTTP connection back to the same server there is the possibility of deadlock under load. To work around the problem, use a separate execute queue for the servlet that returns the DTD.
042556	The <code>weblogic.Admin</code> command now requires the user and password options. For example: <pre>java weblogic.Admin -username system -password gumby1234</pre> The username "system" is required for most functions (for example: VERSION). If you do not specify <code>-username system</code> , you will get the following error: Exception in thread "main" <pre>java.lang.SecurityException: Authentication for user system denied in realm weblogic</pre> <<no stack trace available>>
051980	When using the 1.3 VM with HotSpot, doing a thread dump causes the JVM to exit and the server to crash.
052300	Using Call-by-Reference doesn't work when you deploy your application as an .ear file that is not located in the applications directory and if you have set <code>ServletReloadCheckSecs</code> is set to -1.
052406	Attempting to start Managed Servers in quick succession creates problems. A delay between starting them fixes the problem.
052543	When passing complex objects server to server using RMI-IIOP, there will be many problems using <code>com.sun.jndi.cosnaming.CNContextFactory</code> when getting an initial context; it is recommended to use <code>weblogic.jndi.WLInitialContextFactory</code> instead.
053328	If you are having problems with out of memory errors while using HP, try increasing the kernel parameter called <code>max_thread_proc</code> to 1024.

---

---

Change Request Number	Description
053330	The SNMP agent uses an object identifier (OID) for WebLogic in communicating with SNMP management systems. The WebLogic OID for 6.1 is different than the WebLogic OID for 5.1. In 5.1 it was .1.3.6.1.4.140.600, but in 6.1 it is .1.3.6.1.4.140.625.
054504	We have encountered problems on Windows 2000 and NT when using the <code>-hotspot</code> option with the JVM. This is due to a Sun bug ( <a href="http://developer.java.sun.com/developer/bugParade/bugs/4479571.html">http://developer.java.sun.com/developer/bugParade/bugs/4479571.html</a> ) and results in intermittent crashes of the server. To work around this problem, try setting: <pre>-XX:MaxPermSize=128m</pre> This option allows you to increase the size of the area HotSpot uses for class representations.
057008	The <code>NTSocketMuxer</code> <code>getCPUCount()</code> is returning an incorrect value. A Service Pack from Microsoft that is installed does not recognize all of the CPUs.
058868	Due to a HotSpot virtual machine error, a signal 11 may unexpectedly crash WebLogic Server running on Solaris.
059018, 063937, 063939, 063965	By default, a WebLogic Server 6.1 Service Pack 2 client does <b>not</b> interoperate with a WebLogic Server 6.1 GA or 6.1 Service Pack 1 server. This is true both for a 6.1 SP2 client and a 6.1 SP2 server acting as a client to a 6.1 GA or 6.1 SP1 server. To allow an SP2 client to talk to a 6.1 GA or 6.1 SP1 server, add the following switch to the WebLogic Server 6.1 Service Pack 2 start script: <pre>-Dweblogic.61compat=true</pre>

---

<b>Change Request Number</b>	<b>Description</b>
065410	<p>Due to a Sun bug, a <code>java lang.OutOfMemory</code> error may occur when starting WebLogic Server if the number of classes being loaded is too high.</p> <p>As a workaround, increase the value of the JVM option <code>-XXMaxPermSize</code> when initializing WebLogic Server, using this syntax:</p> <pre>java -XX:MaxPermSize=&lt;value&gt;K</pre> <p>where <code>&lt;value&gt;</code> is some number of kilobytes. The JVM uses a default maximum value of 32 MB.</p> <p>This problem has been reported to Sun.</p>
078289	<p>When a browser using Java Plug-in 1.3.1_02 or 1.3.1_03 refreshes a page with applets, the JVM issues simultaneous <code>destroy()</code> and <code>init()</code> calls to the applets. Because the calls are at the same time, applets may not properly close before trying to initialize.</p> <p>As a workaround, use <code>thread.sleep()</code> in the <code>init()</code> method of your applets to delay initialization by approximately 1000 milliseconds. The delay ensures that the <code>destroy()</code> call finishes any required cleanup before the applet attempts to reinitialize.</p>

# Servlet and Web Application Known Problems

Change Request Number	Description
041528	Cookies generated by the first release of WebLogic Server 6.0 with no Service Pack applied are not recognized by WebLogic Server 6.1.
046536	HTTPS does not work with Virtual Hosting.
046537	Virtual Hosts should define their own security realms.
048678	Previously, to log out a user from a Web Application you could use <code>session.invalidate()</code> . This now only works if the user has been to just one Web Application. If more than one Web Application is visited, <code>session.invalidate()</code> will not log out the user. This is not a known issue; rather it is a design change due to the implementation of Single-Signon. The server tracks the user at the Web server level and not the Web Application level--this behavior is mandated by the Servlet 2.2 specification. If you wish to log out users who have been to more than one Web Application, use: <pre>weblogic.servlet.security.ServletAuthentication. logout(request);</pre>
050811	If you modify a filter class, you must redeploy the Web Application to apply the change.
051311	You cannot not put white space in the name (context path) of a Web Application.

<b>Change Request Number</b>	<b>Description</b>
051329	<p>The following issue is merely informational and will not be fixed or changed because it is the desired functionality.</p> <p>In the WebLogic Server <code>config.xml</code> file, the Path attribute of an archived Web application should consist only of the path to the archive file, but not the archive file name itself. In contrast, the URI attribute of a Web Application Component should consist only of the archive file name. For example:</p> <pre>&lt;Application Name=foo, Path=/apps&gt; &lt;WebAppComponent Name=foo,URI=foo.war&gt; &lt;/Application&gt;</pre> <p>For backward compatibility, WebLogic Server 6.1 accepts a <code>config.xml</code> that specifies both the path and the archive filename of an application in the Path attribute (for example, <code>Path=/apps/foo.war</code>). However, the XML file is overwritten to specify only the path (<code>/apps</code>).</p>
052686	<p>The Web Application Deployment Descriptor Editor shipped with WebLogic Server 6.1 supports Public Final Draft 1 of the Servlet 2.3 specification. Because of this, there is no means within the editor for configuring the <code>&lt;ejb-local-ref&gt;</code> element of <code>web.xml</code>, which was introduced in Public Final Draft 2. Further, the editor will remove any <code>&lt;ejb-local-ref&gt;</code> that may be present in an existing Web App deployment descriptor.</p>
CR088485	<p>WebLogic Server 6.1 SP03 contained the wrong version of the Jakarta regular expression matching library. It was shipped with <code>jakarta-oro-2.jar</code>, rather than <code>jakarta-oro-2.0.6.jar</code>. This resulted in unexpected results for the customer.</p> <p>Work-around: place the correct version of Jakarta—<code>jakarta-oro-2.0.6.jar</code>—in the WebLogic Server classpath, ahead of <code>weblogic.jar</code>.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

089868

An important performance fix introduced in WebLogic Server 6.1 Service Pack 3 (“083654” on page 6-246) caused multi-byte headers to become garbled; previously the default behavior was that Weblogic sent HTTP headers as raw bytes. This previous behavior was incorrect as the HTTP specification states that headers should be ASCII single byte characters. (BEA is aware that this fix can cause interoperability problems between WebLogic Servers 6.1 with Service Pack 3 and lower and WebLogic Servers with Service Pack 4.)

For SP03 and SP04 users experiencing header garbling, the workaround is:

- Install the CR089868\_610sp3.jar patch or CR089868\_610sp4.jar, available from BEA Customer Support, and
- set UseHeaderEncoding="true" in the the <WebServer> element of your config.xml. If set to "true", the server will use the Content-Type character set to encode the Content-Disposition header. Otherwise, and by default, it will use the default platform encoding. Note that the server will not use any encoding if the resulting encoding is an 8-Bit Character set.

In SP05, the workaround for the garbling problem is simply to set UseHeaderEncoding="true" as described in the second bullet item above.

---

# System Administration Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
034177	The Administration Console does not set the correct attribute value for application status.
034955	The application directory list used to scan for deployments is not persistent. Multiple application directories are not supported.
035670	When you reset an MBean name, the configuration is not saved correctly. To work around this problem: <ol style="list-style-type: none"><li>1. In the Administration Console, clone the MBean.</li><li>2. Rename the clone.</li><li>3. Delete the original MBean.</li></ol>
036201	Occasionally, after making modifications in the Administration Console, WebLogic Server cannot be restarted and a Fatal initialization exception Throwable: weblogic.server.ServiceFailureException: Initializing JNDI: javax.naming.ServiceUnavailableException [Root exception is java.net.UnknownHostException: Unknown protocol: 'Protocol: 'unknown''] message is displayed.

---

Change Request Number	Description
038589	<p>If Managed Server(s) are communicating through HTTPS with the Administrative Server, the discovery of Managed Servers does not work. For example, if the Administrative Server is restarted with the <code>-Dweblogic.management.discover=true</code> command line argument and the running managed servers were booted with the <code>-Dweblogic.admin.host=https://&lt;adminhost&gt;:&lt;adminport&gt;</code> argument, then the discovery of managed servers does not work.</p> <p>Workaround: When Managed Servers are booted with an HTTPS connection to the Administrative Server, do not restart the Administrative Server with the <code>-Dweblogic.management.discover=true</code> argument.</p> <p>Instead, do the following:</p> <ol style="list-style-type: none"><li>1. Restart the Administrative server without the discover flag.</li><li>2. Once the Administrative Server is booted, force discovery by the Admin Server by running the following command: <pre>java weblogic.Admin -url &lt;ADMINHOST&gt;:&lt;ADMINPORT&gt; -username system -password &lt;SYSTEM-PASSWORD&gt; INVOKE -mbean "&lt;YOURDOMAIN&gt;:Type=Domain,Name=&lt;YOURDOMAIN&gt;" -method discoverManagedServers</pre></li></ol>
039033	<p>WebLogic Server complains of an error undeploying an application that does not exist.</p>
039311	<p>When you start a managed server that has been cloned, it throws a <code>ConfigurationException</code>. The workaround is to create a server instead of cloning it.</p>
039532	<p>An <code>.ear</code> file can only be undeployed using the Administration Console once it has been exploded under the applications directory. Also, once the application is undeployed, the only way it can be redeployed is through the Administration Console.</p>

<b>Change Request Number</b>	<b>Description</b>
040034	<p>Attempting to start a server as a Managed Server when it is already running as an Administration Server results in the following misleading error:</p> <pre>&lt;Dec 8, 2000 7:44:02 PM PST&gt; &lt;Error&gt; &lt;Configuration Management&gt; &lt;Error connecting to admin server and initializing admin home: admin URL: t3://localhost:7001   java.lang.NullPointerException</pre> <p>Workaround: This is incorrect usage. The same server cannot be started as an Administration Server and a Managed Server. Define a new server for your Managed Server, and specify its name in <code>-Dweblogic.Name</code> during startup.</p>
040124	<p>While the Administration Server is running, copying an existing <code>.jar</code> or <code>.ear</code> file to an applications directory does not redeploy the application. The following error is displayed in the server log:</p> <pre>javax.naming.NameAlreadyBoundException: Can't rebind anything but a replica-aware stub to a name that is currently bound to a replica-aware stub; remaining name ' ' &lt;Dec 12, 2000 12:09:37 PM PST&gt; &lt;Error&gt; &lt;J2EE&gt; &lt;Error deploying application &lt;YOUR-EJB-NAME&gt;: Could not deploy: '&lt;YOUR-FILE-NAME&gt;': JNDI name in use</pre> <p>Make sure that either the <code>guest</code> user or the <code>everyone</code> group is granted lookup permission to <code>weblogic.jndi</code>. The <code>fileRealm.properties</code> file should contain the following:</p> <pre>acl.lookup.weblogic.jndi=everyone</pre>
041290	<p>If the <code>username</code> property is set from the <code>weblogic.Admin</code> or <code>weblogic.Server</code> command lines, the username must be “system”. Note however, that the username syntax is retained to prevent breaking a user’s scripts. The username may not be modified in the <code>config.xml</code> file. The <code>weblogic.management.password</code> property value must be the system password.</p>
042232, 055217	<p>It is no longer possible to have a domain with the same name as its Administration Server. This will not be fixed, as the namespace conflicts produce too many problems.</p>

Change Request Number	Description
048833	The <code>ServletRuntime</code> mbean contains the method <code>getName()</code> . In previous releases, this method returned the name of the servlet. For all other mbeans, the <code>getName()</code> method returns the name of the mbean. For consistency, in WLS 6.1 the <code>getName()</code> method on the <code>ServletRuntime</code> mbean has been modified to return the name of the mbean. A new method, <code>getServletName()</code> , has been added to the <code>ServletRuntime</code> mbean to return the name of the servlet.
052657	If you set configuration attributes for an application in <code>config.xml</code> , yet the application does not really exist, you will see the application incorrectly displayed as deployed in the Administration Console.
077958	<p>When using <code>NodeManager</code>, the following exception may be thrown:</p> <pre>&lt;May 18, 2002 8:53:36 PM PDT&gt; &lt;Error&gt; &lt;NodeManager@localhost:5555&gt; &lt;SocketInputHandler: handshake failed 'FATAL Alert:HANDSHAKE_FAILURE - The handshake handler was unable to negotiate an acceptable set of security parameters.' on socket /172.17.24.148&gt;</pre> <p>This exception could be due to one of the following:</p> <ul style="list-style-type: none"> <li>■ The JVM version being used to run <code>NodeManager</code> is not supported by WebLogic Server.</li> <li>■ A JSEE conflict if <code>NodeManager</code> is running on the AIX platform. If <code>NodeManger</code> is running on the AIX platform, see <a href="#">Weblogic AIX Platform Support</a> in <i>Platforms</i> for more information.</li> </ul>
CR093687	<p>In WebLogic Server 6.1 SP03 and later, you cannot create a Dynamic Connection Pool when <code>LoadBeforeAppDeployments=true</code> for Startup class. The connection pool is not created and goes to a hung state. There is no exception reported.</p> <p>If but, if <code>LoadBeforeAppDeployments=false</code>, the connection pool is successfully created.</p>
CR097152	<p>In WebLogic Server 6.1 SP03 and SP4, restarting the Administration Server may result in this exception:</p> <pre>java.rmi.ConnectException: This RJVM has already been shutdown</pre> <p>This problem was not reproduced in In WebLogic Server 6.1 SP05.</p>

## Tools Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
063745	The documentation on the <code>webllogic.refresh</code> tool states that you can specify files using wildcards (e.g. <code>"*.gif"</code> ) and that you can use a comma-separated list to specify multiple files. However, in WebLogic Server Service Pack 2, <code>webllogic.refresh</code> does not allow the use of wildcards in comma-separated lists.

---

## Web Services Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
046807	The exception messages generated by the Java client application in the Java client JAR file are not internationalized.
047435	In RPC-style Web services, user-defined exceptions thrown by the EJB are not propagated to the client applications that use the Web services client API to invoke the service. Instead, they are propagated as generic SOAPFault exceptions.
049966	<p>If you use the WebLogic Server Web Service client API (version 6.1) in a WebLogic Server 6.0 component, such as an EJB, you get the error <code>java.rmi.ConnectException: No available router to destination</code>. This is because a 6.0 JNDI URL factory gets called rather than the initial context factory that you specify in your client code.</p> <p>To work around this problem, get rid of the 6.0 JNDI URL factories by adding the following code snippet when you make your initial context:</p> <pre>h.put(Context.URL_PKG_PREFIXES, " ");</pre>
051917	In message-style Web services, JMS exceptions that occur on the target JMS destination are not propagated to the client applications that use the Web services client API to invoke the service.

---

Change Request Number	Description
055062	The <code>path</code> attribute of the <code>client.jar</code> element of the <code>wsgen</code> Ant task does not, as documented, default to <code>client.jar</code> if the <code>path</code> attribute is not specified in the <code>build.xml</code> file.
055096	You cannot pass a JavaBean that has no package name as a parameter to a WebLogic Web service.
055596	Interoperability issue. WebLogic Web services do not support SOAP 1.1 multi-reference compound data types.
056287	Interoperability issue. WebLogic Web services do not support the <code>xsd:timeInstant</code> and <code>xsd:dateTime</code> in the SOAP encoding.
056452	Interoperability issue. WebLogic Web services do not support multi-dimensional array data types.
056634	Interoperability issue. WebLogic Web services do not support SOAP-ENV:Header attributes.
058175	Interoperability issue. WebLogic Web services do not support <code>xsd:ur-type</code> array data types.
058840	Interoperability issue. The WebLogic Web service client API cannot bind to a WSDL that specifies more than one return type.
059782	Interoperability issue. The WebLogic Web service client API does not support invoking a Web service whose WSDL contains a <code>&lt;soap:header&gt;</code> element in its <code>&lt;binding&gt;</code> section.
059858	Interoperability issue. WebLogic Web services use only the 1999 XML Schema URL in the generated SOAP request, as shown in the following SOAP request snippet:  <pre>xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance' xmlns:xsd='http://www.w3.org/1999/XMLSchema'</pre>
061161	Interoperability issue. The WebLogic Web services client API does not support the <code>use="literal"</code> attribute of the <code>&lt;soap:body&gt;</code> element.
061335	Interoperability issue. The WebLogic Web services client API does not support invoking a Web service whose WSDL contains a <code>&lt;part&gt;</code> element with an <code>element</code> attribute inside of a <code>&lt;message&gt;</code> element.

<b>Change Request Number</b>	<b>Description</b>
061350	<p>Interoperability issue. By default, the WebLogic Web services client API generates SOAP requests with qualified names. This does not interoperate well with some other non-WLS Web services, such as .NET.</p> <p>The workaround to this problem is to add the following code to your client application:</p> <pre>CodecFactory factory = CodecFactory.newInstance(); SoapEncodingCodec codec = new SoapEncodingCodec(); codec.writeQualifiedNames( false ); factory.register( codec );</pre>
061572	<p>Interoperability issue. The WebLogic Web services client API does not support the <code>style="document"</code> attribute of the <code>&lt;soap:binding&gt;</code> element of WSDL.</p>
061606	<p>When assembling a message-style Web service using the <code>wsgen</code> Ant task, the task adds a literal encoding to the generated WSDL. This forces a programmer invoking the Web service to register the <code>LiteralCodec</code> with the <code>CodecFactory</code>, even when it is not required. The <code>LiteralCodec</code> forces the programmer to have the <code>weblogic.jar</code> file on the client computer.</p>
061929	<p>Interoperability issue. The WebLogic Web services client API always uses qualified names in its generated SOAP requests, even if the client application includes the following code:</p> <pre>codec.writeQualifiedNames( false );</pre>
77911	<p>WebLogic Server correctly maps the built-in Java data type <code>byte[]</code> to <code>xsd:hexBinary</code>. However, the JAX-RPC specification states that <code>byte[]</code> should also have a mapping to <code>xsd:base64Binary</code>. Currently, there is no way to specify this mapping using Version 6.1sp3 of WebLogic Web services.</p>
78062	<p>WebLogic Web services cannot interoperate with .NET Web services that use arrays and JavaBeans as parameters or return values.</p>
78530	<p>The Web Service examples do not build on the Solaris operating system.</p>

# WebLogic Tuxedo Connector Known Problems

---

Change Request Number	Description
052737	<p>A misconfigured node name in the WebLogic Tuxedo Connector XML configuration file will cause the system to hang.</p> <p>Use the following step to restart your system:</p> <ol style="list-style-type: none"><li>1. Shut down WebLogic Server.</li><li>2. Verify your node names in your WebLogic Tuxedo Connector XML configuration file.</li><li>3. Update the WebLogic Tuxedo Connector XML configuration file with valid node names.</li><li>4. Restart WebLogic Server.</li></ol>
077553	<p>Integer array object passed by value from Tuxedo to WebLogic Server throws <code>weblogic.utils.AssertionError</code>.</p> <p>This problem was resolved in WebLogic Server 6.1 SP04. See "<a href="#">077553</a>".</p>
078774	<p>A tBridge session connection may stop pulling messages from the Tuxedo /Q.</p>

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

079630

Most users will expect to set the Correlation ID using the `setJMSCorrelationID(String)` method. This will take a 32 character string and turn it into a 64 byte array. JMS stores the string as UTF-16BE. When the `tBridge` receives the correlation id from Tuxedo it is 32 characters as 32 bytes. The `tBridge` then uses the `setJMSCorrelationIDAsBytes(byte[])` method to set the message for JMS receiving queue. The characters may be the same in ASCII but the two will not compare due the length difference.

When it is necessary to compare the Correlation ID with the ID returned from Tuxedo in the receiving JMS queue, create a byte array containing the hexadecimal values of the Correlation ID. Then use the `setJMSCorrelationIDAsBytes()` and `getJMSCorrelationIDAsBytes()` methods to create an ID to compare with the ID returned from Tuxedo in the receiving JMS queue.

For example, if the string Correlation ID is

"1234567890ABCDEFGHIJKLMNQRSTUUV":

```
private byte[] coridbyte={0x31,0x32,0x33,0x34,0x35,0x36,0x37,  
                          0x38,0x39,0x30,0x41,0x42,0x43,0x44,  
                          0x45,0x46,0x47,0x48,0x49,0x4a,0x4b,  
                          0x4c,0x4d,0x4e,0x4f,0x50,0x51,0x52,  
                          0x53,0x54,0x55,0x56};  
msg.setJMSCorrelationIDAsBytes(coridbyte);  
corIDasBytes = msg.getJMSCorrelationIDAsBytes();
```

`corIDasBytes` contains the correct value to compare with the Correlation ID returned from Tuxedo.

---

---

# XML Known Problems

---

Change Request Number	Description
050274	<p>When you use the Administration Console to add either a parser or entity XML Registry entry, and you do not specify a public ID, you cannot subsequently navigate to the entry to edit it.</p> <p>The workaround is to always enter a value in the public ID field when you create an XML registry entry.</p>
052985	<p>The built-in Xalan transformer package <code>weblogic.apache.xalan</code> does not include the Xalan compatibility API. These classes are included in the <code>xalanj1compat.jar</code> file if you were to download Xalan 2.0.1 directly from the Apache Web site. For more information on the compatibility API, see <a href="http://xml.apache.org/xalan-j/usagepatterns.html#compat">http://xml.apache.org/xalan-j/usagepatterns.html#compat</a>.</p> <p>Because these classes are not included in WebLogic Server, if your application uses code from Xalan 1.X that has been deprecated in Xalan 2.0.1, your application will not work correctly. Modify your code so that it does not use the Xalan 1.x deprecated classes, or rewrite your Xalan code to use JAXP instead.</p>

---

# ZAC Known Problems

---

<b>Change Request Number</b>	<b>Description</b>
CR047534	<p>The JRE Update Frequency property set in the ZAC Publishing Wizard is no longer supported, and you need to use one of the following methods to update your JRE when necessary.</p> <p>When a JRE is packaged with an application, it will be reinstalled on the client only if:</p> <ul style="list-style-type: none"><li>■ The entire JRE installation directory is deleted from the client AND either the client executable OR the ZAC bootstrap routine is run; or</li><li>■ A new version of the JRE library package is published on the WebLogic Server AND the bootstrap routine (not the client executable) is run from the client.</li></ul> <p>No package versioning takes place on the server (unless the version number is made part of the package name); it is only possible to "revert" a package if it has not already been published.</p>
061699	<p>WebLogic Server does not automatically register a default web application. Because ZAC gets classes through the default web application, users of ZAC must now explicitly configure a default web application.</p> <p>If you fail to explicitly configure a default web application, ZAC will not function, and the server log will contain a message similar to the following:</p> <pre>&lt;Nov 2, 2001 2:50:49 PM PST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;HttpServer(7908535,null default ctx,myserver) found no context for "/drp-exports/ZACHello/index.xml". This request does not match the context path for any installed web applications and there is no default web application configured.&gt; &lt;Nov 2, 2001 2:50:49 PM PST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;HttpServer(7908535,null default ctx,myserver) found no context for "/drp-publish/ZACHello/". This request does not match the context path for any installed web applications and there is no default web application configured.&gt;</pre>

---

# 6 Resolved Problems

The following sections describe problems that have been resolved by Service Packs for WebLogic Server 6.1. Service Packs are cumulative; the current release, Service Pack 7 contains all the fixes made in earlier Service Packs released for WebLogic Server 6.1.

- WebLogic Server 6.1 Service Pack 7 Solutions
- WebLogic Server 6.1 Service Pack 6 Solutions
- WebLogic Server 6.1 Service Pack 5 Solutions
- WebLogic Server 6.1 Service Pack 4 Solutions
- WebLogic Server 6.1 Service Pack 3 Solutions
- WebLogic Server 6.1 Service Pack 2 Solutions
- WebLogic Server 6.1 Service Pack 1 Solutions
- WebLogic Server 6.1 Release Solutions

## WebLogic Server 6.1 Service Pack 7 Solutions

The following sections describe problems that have been resolved for the release of WebLogic Server 6.1 Service Pack 7.

- Administration Console

- Classloader
- Cluster
- Core
- Deployment
- EJB
- Internationalization
- JDriver
- JDBC
- JMS
- JNDI
- JSPs and Servlets
- JVM
- Operations, Administration and Management
- Plug-Ins
- RMI-IIOP
- RMI
- Samples
- Security
- SNMP
- WLEC
- WebLogic Tuxedo Connector

## Administration Console

CR Number	Description
CR172370	<p>When the Administration Console created an Application MBean, the associated WebAppComponent MBean incorrectly inherited the Application MBean name instead of the WebAppComponent's URI name. This caused the deployer tool to incorrectly create an additional deployment repository for this application when it was updated.</p> <p>This defect was corrected by a code change.</p>
CR105735	<p>Navigation through the directories did not go through action servlet, so encoding was not set properly.</p> <p>A change was made to set the encoding explicitly in the page, so the correct encoding is used to produce the response.</p>
CR087598	<p>Propagation from list page to realm deletion changed the MBean class name specific to the MBean to be deleted. Because of this MBean showed only the last deleted type MBeans in the list.</p> <p>A attribute (original class name from list) has been introduced which will be propagated in the above described flow. This attribute provides a way to recover the original class name and listing remains undisturbed. This solution applies to all MBean types.</p>
CR184891	<p>The console jsp that was used to upload applications to remote servers was calling the <code>ApplicationManager.update()</code> method even when the server was not in development mode.</p> <p>Removed the call to <code>ApplicationManager.update()</code> if <code>ProductionMode</code> is enabled. Uploaded applications are no longer automatically deployed when <code>Production Mode</code> is enabled.</p>
CR097378	<p>A <code>NullPointerException</code> sometimes appeared on the Administration Console while the <code>WebApplication Deployment Descriptor</code> was being edited.</p> <p>WebLogic Server was not checking for the descriptor being null, which is the case when it can not be parsed. For this reason a <code>NullPointerException</code> was thrown.</p> <p>If the descriptor is null, the proper error action will now be called to display the correct message.</p>

<b>CR Number</b>	<b>Description</b>
CR097036	<p>The correct icons were not being retrieved by the console pages due to changes in the <code>getServletContext</code> code to be in compliance with the Servlet 2.3 specification.</p> <p>Code changes were made to <code>buildIconList()</code> to provide the path to the images subdirectory in the <code>console.war</code>.</p> <p>Correct icons now appear in the navigation applet.</p>
CR044503	<p>Creating a new ACL with a blank Permission caused an unexpected error. This was due to names not being checked for null and blank values under some circumstances.</p> <p>Appropriate checks have been added, and messages for null names have been corrected.</p>
CR08296	<p>The <code>getParameter()</code> method in <code>GraphApplet</code> is used to get the value of the min/max heap size. Although the return type is long, WebLogic Server was using <code>Integer.parseInt</code> to convert the string values. When the heap size was over 2G, there was an overflow in the integer value. Because of this, the current free memory was sometimes reported as a negative number.</p> <p>WebLogic Server now uses <code>Long.parseLong</code> to avoid overflow and to get the correct value of the heap even if it is over 2G .</p>
CR079771	<p>WebLogic Server was sending a file link to the browser to report the results of start cluster and start/kill domain operations. That link worked only if the browser was running on the same machine as the administration server. This occurred on all platforms.</p> <p>WebLogic Server now sends a action link which will process the file and send the output to the browser in an HTML format.</p>
CR077058	<p>When a <code>JMSJDBCStore</code> was created, the "None" option was eliminated from the Connection Pool drop down list.</p> <p>Adding the ability to make <code>JDBCConnectionPool</code> null restored "None" to the list.</p>
CR069130	<p>When an application with an incorrect <code>web.xml</code> file was added to the applications directory, the Administration Console reported it as having been deployed even though it was not deployed at startup.</p> <p>WebLogic Server now correctly reports such applications as undeployed.</p>

<b>CR Number</b>	<b>Description</b>
CR075168	<p>WebLogic Server was using the greater than symbol, '&gt;', as the Token delimiter. Because of this, if a custom message contained a '&gt;', it was being truncated while being displayed.</p> <p>WebLogic Server no longer truncates the last token when writing to the log.</p>

## ClassLoader

---

<b>CR Number</b>	<b>Description</b>
CR128510	<p>readClassDescriptor() of MsgAbbrevInputStream was trying to resolve a class and throwing a ClassNotFoundException for unknown classes. Java serialization will skip this ClassNotFoundException if corresponding data is not being read.</p> <p>readClassDescriptor() of MsgAbbrevInputStream no longer tries to resolve unknown classes and MsgAbbrevInputStream implemented resolveClass().</p>

---

## Cluster

---

<b>CR Number</b>	<b>Description</b>
CR129234 CR189793	<p>ReplicatedSessionContext has two hashtables, one of which stores the sessions for which the server is primary, and the other stores sessions for which the server is secondary. The hashtable with secondary sessions has sessionId as a key and ROID as value.</p> <p>When a session was invalidated and the request to remove the session came to the secondary server, WebLogic Server passed the ROID to the ReplicatedSessionContext, iterated over the values in the hashtable in order to compare ROID with that value, and then removed the entry. WebLogic Server now avoids iterating over the hashtable by passing the sessionId instead of ROID.</p>
CR135131	<p>weblogic.cluster.MulticastObjectListener was locked on a call to objectAdded and waited to lock a HashMap while in a different thread MulticastReceiver locks the HashMap and attempted a lock on MulticastObjectListener. This caused a Java Level Deadlock.</p> <p>Changing the lock order eliminated the deadlock.</p>

---

CR Number	Description
CR127765	<p>When a node in cluster was restarted, it first tried to synchronize with other cluster members resulting in one server adding the other to the cluster view.</p> <p>Before that cluster node opened the listen port, if a request came in with this cluster node as a secondary, it resulted in an error and from that point on all requests to create secondaries on that cluster node failed.</p> <p>Now during cluster synchronization, when a node is coming up, it first identifies all the running nodes in a cluster and then sends out a broadcast identifying itself. This ensures that the cluster node coming up is identified by other nodes after it opens a listen port.</p>
CR127643	<p>When a Dynamic Proxy that implemented interfaces declared inside the web application was put into the HttpSession and the session was replicatable, WebLogic Server was not able to load the interface classes on the secondary server.</p> <p>Dynamic Proxies implementing interfaces stored in the application archive can now be put into HttpSession and be correctly replicated.</p>
CR111029	<p>The cluster members timed out if they did not receive a heartbeat within the default 30 second timeout period. Heartbeats were sent every 10 seconds and servers waited for 3 periods (total wait time was 30 seconds) to get a heartbeat before the cluster member was timed out and declared unavailable. For example, during session replication if the secondary server was unavailable at TCP level, the 30-second period was sometimes too long for a very busy web site. Before the secondary was removed from the cluster view, the primary tried to replicate many sessions to the secondary and thus caused the server to hang or made the server slower.</p> <p>The timeout value IdlePeriodsUntilTimeout is now tunable. It is set on the &lt;Server IdlePeriodsUntilTimeout="3"&gt; tag in the config.xml file. In general customers should not tune this value and should leave it at the default (3). However, in certain cases depending on the load, available redundancy in the architecture and specific application problems and/or certain production scenarios, tuning this value carefully might alleviate the problem temporarily until the root cause is identified and fixed.</p> <p>BEA WebLogic recommends that you use HIGH caution when changing this value and ensure sufficient testing of your application at peak load scenarios to ensure the expected behavior. There is no recommendation that fits all scenarios, so testing for load and stress is a must if this value needs to be changed.</p>

---

## Core

<b>CR Number</b>	<b>Description</b>
CR190507	Fixed a problem with oneway calls for ReplicationManager..
CR190417	<p>When a request landed on a server that was neither primary nor secondary, it got the session from the existing secondary by looking it up on the secondary server with the host port information from the session id. After it successfully got the session from existing secondary it removed the existing session and tried to create a new session.</p> <p>When two such requests occurred at the same time a distributed deadlock could occur since these requests landed on the 'Replication' thread queue and the 'Replication' thread queue had only two threads and there was no other thread available for reading the response.</p> <p>Code was added to WebLogic Server which fixed the problem so that it will not deadlock.</p>
CR183210	<p>In WebLogic Server 7.0 and higher, the <code>javax.ejb.TransactionRolledbackLocalException</code>'s SVUID was changed and when WebLogic Server 7.0 or 8.1 experienced a <code>javax.ejb.TransactionRolledbackLocalException</code> when a WebLogic Server 6.1 client made a remote call to ejb hosting by a WebLogic Server 7.0 or 8.1 server instance, it experienced an <code>InvalidClassException</code>.</p> <p>This problem is fixed in WebLogic Server 6.1 so that it now understands the incoming <code>final-ejb20 ClassDescriptor</code> for <code>javax.ejb.TransactionRolledbackLocalException</code>.</p>
CR133631	<p>Stopping a windows service configured with <code>beasvc.exe</code> sometimes caused a timeout when a <code>stopclass</code> was specified. The service was timing out because there was a race condition between the <code>stop</code> and <code>main</code> threads of <code>beasvc.exe</code>.</p> <p>The race condition has been corrected and the windows service no longer times out.</p>
CR121483	<p>A <code>ConcurrentModificationException</code> was thrown when the monitoring subsystem was trying to read the values of the <code>abbrev</code> table and at the same time the <code>abbrev</code> table was being modified by the <code>rjvm</code> layer.</p> <p>Cloning the key set before sending the data to the monitoring subsystem so that the <code>HashMap</code> is not modified simultaneously by two threads eliminates the <code>ConcurrentModificationException</code>.</p>

CR Number	Description
CR188371	<p>When an application cached a stateless session bean remote stub, and all the servers in the cluster were restarted, the stub was unable to refresh its lists of server nodes where the remote object was available and failover did not succeed. This was happening because the stub did not have the information needed to re-establish the initial context with the cluster nodes, hence the remote method invocation failed.</p> <p>WebLogic Server code was not propagating the thread environment for the stateless session bean stubs in WebLogic Server 6.1 versions and it is required in WebLogic Server 7.0 and higher versions for unmarshalling to set the environment so failover works.</p> <p>The runtime descriptors for the clusterable stateless session beans now have the propagate-environment attribute set to true by default.</p> <p>The descriptor is now read and the propagateEnvironment is set so the environment is passed on during unmarshalling. This will allow the failover logic to reconnect to the cluster nodes to retrieve the new list of server nodes where the remote object is available and allow for proper failover.</p>
CR182684	<p>Every time the beasvc service handler was called, the beasvc log added a line indicating that it was called.</p> <p>For example:</p> <pre>Tue Apr 27 11:52:17 2004] [I] [service_ctrl] 4</pre> <p>This caused the log to fill up when there was no real activity on the server.</p> <p>The debug statement was removed, eliminating the log problem.</p>
CR181986	<p>WebLogic Server running as a service sometimes ran out of memory if it was using a large number of threads.</p> <p>Reducing the reserve stack size used by beasvc.exe and beasvc64.exe from 1mb to 256kb eliminated the memory problem.</p>
CR176614	<p>Stateful Session Bean handles taken from the 70 (or 81) WebLogic Server servers to WebLogic Server 61 server/client were failing with a ClassNotFoundException in the serialization code.</p> <p>Compared to the WebLogic Server 6.1 server, WebLogic Server 70 and 81 servers used a different PK mechanism for SFSB in the ejb container.</p> <p>Adding the necessary classes to 6.1 line to support the inter-op requirement eliminated the ClassNotFoundException.</p>

## 6 Resolved Problems

---

CR Number	Description
CR174605	<p>According to the documentation in "Starting and Stopping Servers" if the CLASSPATH is too long, it can be added as a single line to a file and then accessed as -classpath @filename. However, when beasvc attempted to load the contents of CLASSPATH file, it truncated the last character when the file did not end with a new line.</p> <p>Now beasvc determines whether the file terminates properly and then reads the file accordingly.</p>
CR173958	<p>Interoperability between 6.1 SP04 and 8.1 SP02 us t3 was failing when the protocol was changed from "secure" to "non-secure" between the front-end and back-end. The front-end QOS was being propagated to the back-end for the authentication call and it was failing.</p> <p>The problem was resolved by using "anonymous" when doing the bootstrap authenticate call.</p>
CR172366	<p>Messages printed by beasvc.exe to the event log were not readable in Japanese locale.</p> <p>A code fix ensures that English messages are printed for all non-English locales.</p>
CR135225	<p>A deadlock was occurring between RJVM and NTSocketMuxer.</p> <p>Code was added to ensure that WebLogic Server does not hold a lock on IORecord during dispatch, thus ensuring that a deadlock will not occur.</p>
CR134971	<p>When &lt;&lt;no stack trace available&gt;&gt; was sent out as part of the exception message field (from the server side) rmi layer was recursively adding / appending the exception as server side stack trace. This was filling up the log files. This manifested when the server was running out of heap and there was a NullPointerException thrown by the application code (EJB + Servlets)</p> <p>WebLogic Server now parses the exception message field, traps this specific exception and ensures that it is appended only once.</p>
CR133880	<p>Under heavy load, T3File client often threw an ArrayIndexOutOfBoundsException when closing OutputStream/InputStream.</p> <p>WebLogic Server now acquires the lock on the Vector before starting the loop.</p>
CR175607	<p>Installing WebLogic Server as a Windows service immediately after uninstalling it sometimes created wrong registry keys, which could lead to startup problems.</p> <p>A code fix ensure that the registry keys are flushed and properly closed.</p>

CR Number	Description
CR130409	<p>When setting the maximum length of an execute queue with the <code>-Dweblogic.kernel.allowQueueThrottling</code> flag to throttle "slow moving resource intensive" requests on a custom queue, clients did not receive a 503 response and therefore waited for the timeout.</p> <p>This problem was resolved by a code fix to check for the dispatch return value on the caller and use the <code>sendError()</code> API to return the 503 response.</p>
CR130376	<p>According to the documentation at "Starting and Stopping Servers" if the <code>CLASSPATH</code> is too long, it could be added as a single line to a file and then accessed as <code>-classpath @filename</code>. However, this was not working because when <code>beasvc</code> attempted to load the contents of <code>CLASSPATH</code> file, it sometimes truncated the last character. This only happened when the file did not end with a new line.</p> <p>A code change was made so <code>beasvc</code> figures out whether the file terminates properly and is then read accordingly."</p>
CR129094	<p>Performance issues that involved TCP window shrinkage in the t3 protocol on the AIX platform have been resolved.</p>
CR122939	<p>A distributed deadlock occurred due to a failure to maintain session stickiness with the primary server. When a request lands on the server which is neither primary nor secondary, it will try to remove the existing session from the primary as well as the secondary and create a new one on the current server and register a new secondary to it. In doing so, this server makes a remote request to the primary to remove the session on non-blocking queue, the primary server makes a remove call to the secondary server to remove itself on non-blocking queue. If there are more than one such requests, there will be no other thread to receive the response on the current server since there will be only two threads available in non-blocking queue and hence there is a distributed deadlock.</p> <p>WebLogic Server now makes no new remote requests while processing requests that come in on a non-blocking queue. This fixes the problem, as it ensures that there will always be a thread available in the non-blocking queue to receive a response.</p>

## Deployment

---

<b>CR Number</b>	<b>Description</b>
CR134122	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_57.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_57.00.jsp</a> .

---

## EJB

---

<b>CR Number</b>	<b>Description</b>
CR124991	When <code>&lt;validate-db-schema-with&gt;MetaData&lt;/validate-db-schema-with&gt;</code> was used AND a <code>cmp</code> -field and a <code>cmr</code> -field were mapped to the same column, a <code>java.lang.IndexOutOfBoundsException</code> was being thrown.  WebLogic Server has been modified and this exception is no longer thrown under these circumstances.
CR055396	When a EJB QL syntax error occurred, WebLogic Server generated an error message with an incorrect xml file reference.  WebLogic Server now generates the message as follows if there are syntax errors in EJB QL.  [java] ERROR: Error from ejbc: Error while reading 'META-INF/ejb-jar.xml' or 'META-INF/weblogic-cmp-rdbms-jar.xml'. The error was:

---

CR Number	Description
CR185643	<p>For BLOBs, in the generated code, WebLogic Server calls <code>ObjectOutputStream.writeObject</code> and <code>ObjectInputStream.readObject</code> to serialize/deserialize the object before writing/reading it to the database. These calls add extra header information. The <code>writeObject</code> method writes the class of the object, the signature of the class, the values of the non-transient and non-static fields of the class, and all of its supertypes are written. (This is the reason for the extra header seen in the database.) These calls do not cause a problem when customers are using only WebLogic Server to set and get BLOBs, because it uses <code>readObject</code> to convert the bytes into the appropriate object, which needs that extra header information. However, if the BLOB has been inserted directly into the database by some other vendor or programmer using:</p> <pre>OutputStream os = ((weblogic.jdbc.common.OracleBlob) lob).getBinaryOutputStream(); os.write(this.tiffImage); // byte[] tiffImage</pre> <p>then problems may occur because WebLogic Server uses <code>readObject</code> and the header information is missing. For the data inserted using WebLogic Server, the other programs that would read the bytes directly get the extra header information and fail. A system property has been added which allows WebLogic Server to correctly report header information.</p>
CR187121	<p>A high value for <code>idleTimeoutSecs</code>, for instance, 60000000, in the Deployment Descriptor when multiplied by 1000 to convert it into msec was overflowing into a negative value. This caused the trigger that cleans the passivated beans from the disk to constantly fire, causing high CPU usage.</p> <p>The variables within the EJB container which held the timeout values in milliseconds, such as <code>idleTimeoutMS</code>, <code>sessionTimeoutMS</code>, and <code>readTimeoutMS</code>, have been changed from the <code>int</code> type to <code>long</code>. This prevents any numeric overflow.</p>
CR135367	<p><code>EJBStatelessHomeRuntimeMBean</code>'s implementation class did not provide correct details of pool statistics.</p> <p>Code was changed to provide the correct pool statistics in the runtime mbean methods. <code>MBean</code> calls now show the correct data.</p>
CR127097	<p>The Stateful EJB monitoring page did not have the machine name prefixed to its output</p> <p>so it was impossible to see to which machine the entry referred.</p> <p>The machine name was added to the Stateful EJB monitoring page.</p>

## 6 *Resolved Problems*

---

<b>CR Number</b>	<b>Description</b>
CR124026	<p>The read-only concurrency strategy in 6.x used Exclusive concurrency. This guaranteed exclusive access to the bean, but not to the generated Set for the 1-N relationship. It was possible for two clients to get access to the same Set. In such a case, when some method was called on the relationship Set, it caused a <code>NullPointerException</code>.</p> <p>Relevant code changes were made to guarantee exclusive access to the relationship Set to fix the issue.</p>
CR104539 CR102308	<p>In WebLogic Server 6.1 SP04, the Administration Console reported incorrect values for waiters for entity beans.</p> <p>The problem was solved by adding a <code>waiterCurrentCount</code> attribute that is incremented when a client starts waiting for a lock and decremented when the lock is acquired or the client times out.</p>
CR096398	<p><code>EJBContext.getRollbackOnly()</code> returned "true" if a transaction marked for rollback and had not yet been rolled back. After the transaction was rolled back, it returned "false".</p> <p><code>EJBContext.getRollbackOnly()</code> now returns "true" even if the transaction has already been rolled back.</p>
CR060229	<p>The Administration Console was not exposing the Transactions Committed Total Count, the Transactions Rolled Back Total Count or the Transactions Timed Out Total Count for Stateful and Entity EJBs.</p> <p>The Administration Console now allows monitoring of these items.</p>
CR087261	<p>The <code>EJBDeployer</code> was writing an incorrect deployment message to the log for Message Driven Beans.</p> <p>The correct message is now being logged when a Message Driven Bean is deployed.</p>

## Internationalization

---

CR Number	Description
CR079432	<p>MessageLocalizer was not setting the l10n_package attribute in localized catalog files using the l10ngen utility.</p> <p>MessageLocalizer now correctly sets the l10n_package attribute in a localized catalog file.</p>

---

## JDriver

---

CR Number	Description
CR142730	<p>After a long database outage, a JDBC connection pool that used the XA jDriver for Oracle with TestConnectionsOnReserve="true" could not recover and recreate connections to the database. The following error messages were thrown:</p> <p>&lt;Warning&gt; &lt;JDBC&gt; &lt;001096&gt; &lt;Refreshing this bad pool connection failed weblogic.common.ResourceException: java.sql.SQLException: open failed for XAResource 'oracleXAPool' with error XAER_RMERR: A resource manager error has occurred in the transaction branch.</p> <p>and</p> <p>&lt;Warning&gt; &lt;JDBC&gt; &lt;001096&gt; &lt;Refreshing this bad pool connection failed weblogic.common.ResourceException: java.sql.SQLException: LDA pool exhausted - make sure you call Connection.close()</p> <p>The problem was that during the outage, the JDBC connection pool attempted to recreate connections, but on failure, those connection attempts were not cleaned up and were depleting Oracle client resources (in the LDA).</p> <p>The jDriver now cleans up connection creation attempts that fail.</p>

---

## 6 *Resolved Problems*

---

<b>CR Number</b>	<b>Description</b>
CR134285	<p>class weblogic.db.oci.OciLob defined two class level variables to hold the returned byteArray for either BLOB or CLOB data from database. As this byte array could be quite large each instance of OciLob filled up the heap quickly until a garbage collection occurred.</p> <p>Further testing and a modified OciLob moving the class level variables retBArray[] and retCArray[] into the methods caused the variable to be allocated on the stack and reduced the size of the OciLob object instance and thus overall heap usage was reduced.</p> <p>Code was added to move the global variables retBArray and retCArray to the method level in order to reduce the size of the memory heap usage.</p>
CR172462	<p>The WebLogic Server jDriver was not functioning properly with Oracle 9.2 when using the AL32UTF8 character set.</p> <p>This problem was resolved with a code fix.</p>
CR136168	<p>WebLogic jDriver may cause the server to crash with Oracle error message ORA-02392 when using the long raw type under heavy loads.</p> <p>A code fix was implemented to resolve this issue.</p>
CR129220	<p>WebLogic Server Oracle jDriver was not properly releasing Clob Objects for garbage collection.</p> <p>WebLogic Server now releases Clob Objects correctly.</p>

## JDBC

CR Number	Description
CR095876	<p>In previous releases, methods in the <code>weblogic.jdbc.common.Pool</code> and <code>weblogic.jdbc.common.JDBCServices</code> interfaces were referenced in the Programming WebLogic JDBC guide, but the Javadocs for those interfaces were not published.</p> <p>In WebLogic Server 6.1 SP7, the Javadocs for these interfaces are published at <a href="http://e-docs.bea.com/wls/docs61/javadocs/weblogic/jdbc/common/package-summary.html">http://e-docs.bea.com/wls/docs61/javadocs/weblogic/jdbc/common/package-summary.html</a>. Note, however, that the methods in these interfaces have been deprecated and are not available in WebLogic Server 7.0 and 8.1.</p>
CR128888	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_53.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_53.00.jsp</a>.</p>
CR127949	<p><code>Statement.getResultSet()</code> sometimes generated an unnecessary new <code>ResultSet</code> wrapper for the one underlying DBMS resultset, if a result set had already been returned to the user code. If the user code had run an <code>executeQuery()</code> call first, without retaining the result set it returned, garbage-collecting could close it immediately, with the underlying DBMS resultset, invalidating and prematurely closing any result set returned by a subsequent <code>getResultSet()</code> call.</p> <p>WebLogic Server no longer generates unnecessary wrappers due to a code change.</p>
CR184143	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_53.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_53.00.jsp</a>.</p>
CR182410	<p>Under a high load condition, WebLogic Server slowed significantly.</p> <p>Removing synchronization In <code>weblogic.jdbc.common.internal.MultiPool.searchLoadBalance()</code> method eliminated the slowdown under high load conditions.</p>
CR133612	<p>In Service Pack 7, the Oracle thin driver is modified with latest 9.2.0.4 drivers in classes12.zip in the <code>3rdparty/oracle/920</code> directory</p>
CR131575	<p>WebLogic Server sometimes threw an incorrect warning about a JDBC connection leak that began:</p> <pre>[SerialConnection]: Connection Leak detected!!!!!!java.lang.Throwable: StackTrace at creation of connection: /n</pre> <p>The leak detection code that sent this warning is obsolete. A code change resolved the problem.</p>

<b>CR Number</b>	<b>Description</b>
CR130306, CR135909	<p>In <code>jta.DataSource</code>, when doing <code>refreshAndEnlist</code>, WebLogic Server called <code>tx.enlist()</code>, but the connection was not returned to the pool if there was an exception in the <code>refreshAndEnlist</code> call.</p> <p>A code change catches the exception and releases the connection to the pool.</p>
CR129379	<p>When an EJB transaction created many new entities or otherwise engaged many beans that all use JDBC, WebLogic Server risked running out of Oracle cursors, because in an attempt to avoid a suspected Oracle driver bug, WebLogic Server delayed closing JDBC statements until the end of a transaction, holding the cursors for the statement until then.</p> <p>This behavior has been changed so that the session need not hold cursors until the transaction ends.</p>
CR127720	<p>New versions of JDBC drivers track the transactional state of connections. If a local transaction was active on a connection, XA operations could not be performed on it, resulting in an <code>XAER_PROTO</code> or <code>XAER_RMERR</code> when an <code>xa_start()</code> was called on the connection. As a result, applications had to go through the tedious process of narrowing down where in their code they had started but not ended a local transaction.</p> <p>The problem was resolved by a code change in the recovery method that prevents special XA connections from being released to the pool twice.</p>

---

# JMS

CR Number	Description
CR188040	<p>WebLogic JMS was not receiving notification when <code>ServerDebugMBean()</code> JMS attributes were changed.</p> <p>The <code>ServerDebugMBean()</code> can now be used to dynamically enable and disable "DebugJMS" <code>ServerDebugMBean</code> attributes. This allows customers to dynamically enabled or disable JMS Debugging flags.</p>
CR099554	<p>Under certain circumstances when a server with JMS messages in a pending state was shut down or crashed, pending messages were not recovered when the server was restarted.</p> <p>Pending messages are now recovered upon restarting the server.</p>
CR123194	<p>In previous WebLogic Server 6.1 service packs, when the server instance went down but its clients remained active, JMS threw a runtime exception <code>weblogic.rmi.extensions.RemoteRuntimeException</code>, instead of a <code>JMSException</code> as expected per the JMS specifications.</p> <p>A code change has resolved the problem in this Service Pack.</p>
CR126192	<p>Long-lived JMS connections lacked a periodic heartbeat check.</p> <p>Following a code change, when JMS is idle the connection pings the database every five minutes to keep connection fresh.</p>
CR182338	<p>A receiver stops processing messages when a <code>RedeliveryLimit</code> is configured and the number of times the <code>RedeliveryLimit</code> is reached is greater than the <code>MessagesMaximum</code> setting on the <code>ConnectionFactory</code>.</p> <p>Example: <code>RedeliveryLimit=0</code>, <code>MessagesMaximum=10</code>. Receiver receives a message and calls <code>session.recover()</code> 10 times. Receiver will stop processing messages and the console will show 10 messages pending.</p> <p>Code was added to adjust the window counter when a message is removed from the queue because the <code>RedeliveryLimit</code> had been reached.</p>
CR126183	<p>An idle bridge was logging a message after the maximum idle time setting had been reached.</p> <p>Code was added to suppress the repetitive log message "Bridge X start transferring messages" logged by an idle bridge.</p> <p>If the bridge is stopped and restarted, or if it encounters an exception and is restarted you will see the "Bridge "bridgename" starts transferring messages" log message, but you will not see the repetitive message logged by an idle bridge.</p>

## **6** *Resolved Problems*

---

---

<b>CR Number</b>	<b>Description</b>
CR133155	WebLogic Server took too long to recover JMS messages from the JDBC store at boot time. Including JMS in the getTables prefix resolved the problem.

---

## JNDI

CR Number	Description
CR136746	<p>Enhancement request to allow explicit naming of JDBC driver being used in class VendorId.</p> <p>This customer enhancement was made through a code change.</p>

## JSPs and Servlets

CR Number	Description
CR122556	<p>HTTP tunneling requests were sometimes producing ProtocolExceptions. Code was added to WebLogic Server to eliminate the ProtocolExceptions.</p>
CR087857	<p>When the character encoding was set by JSP pages or servlets, the ServletOutputStream.write(int) method, which takes int type as its argument, received the data encoded using the specified charset encoder.</p> <p>WebLogic Server no longer encodes the binary data when ServletOutputStream.write(int) is called.</p>
CR180425	<p>WebLogic Server was sending wsproxy specific headers even when the request did not originate with a proxy.</p> <p>Code was added to check if the request is coming from a proxy and send the appropriate header.</p>
CR077922 CR063304	<p>The HttpSessionBindingListener were not getting fired correctly. In some cases they were invoked twice.</p> <p>Re-implementing the callbacks strictly per the Servlet Specification fixed this problem.</p>
CR123308	<p>T3ServicesDef and LogServicesDef interfaces have been deprecated beginning with the WebLogic Server 6.1 SP5 release.</p>
CR101992	<p>A web application that had a local EJBObject reference in its session, sometimes got an javax.ejb.EJBException after it was redeployed.</p> <p>Code was added to catch the exception and log a message. An error message is now logged to indicate serialization failure and the getAttribute() of HttpSession, ServletRequest or ServletContext returns null under these circumstances.</p>

<b>CR Number</b>	<b>Description</b>
CR085091	<p>There was a problem that was preventing serving custom error pages, if the request was a conditional GET (Is-Modified-Since header set) for a protected resource.</p> <p>The logic was fixed to serve the custom error page if one is defined.</p>
CR127959	<p>Code that used a response wrapper and called <code>getOutputStream</code> on the original response passed into a JSP resulted in <code>NestedBodyResponse</code> always throwing an <code>IllegalStateException</code>.</p> <p>Mixed use of <code>getWriter()</code> and <code>getOutputStream()</code> from <code>NestedBodyResponse</code> are now allowed, so the exception will only be thrown when appropriate.</p>
CR133291	<p>A protocol exception, <code>excjava.net.ProtocolException: Didn't meet stated Content-Length</code>, was occurring when a client cancelled a request while the default <code>fileServlet</code> was sending a file.</p> <p>This was resolved with a code change.</p>
CR127708	<p>WebLogic Server decoded the path <code>/foo/bar%c0%baz/</code> to <code>/foo/bar</code> because <code>sun.io.ByteToCharConverter</code> stopped converting the remaining bytes if it encountered bytes that were not valid in UTF8 (for example, <code>0xc0</code>). This problem does not occur on JDK 1.4.</p> <p>The problem was resolved with a WebLogic Server UTF-8 converter that replaces invalid UTF-8 sequences with <code>U+FFFD</code> characters. As a result of this fix, the path <code>/foo/bar%c0%baz/</code> is decoded to <code>/foo/bar?%baz</code>.</p>
CR143448	<p>The <code>java.lang.IllegalStateException: HttpSession is invalid</code> exception occurs in the servlet container's internal call. If other threads using the same session ID invalidate the session object during processing of <code>ServletRequestImpl.syncSession()</code>, an <code>IllegalStateException</code> may occur while calling <code>SessionData.putValue()</code> or <code>SessionData.isNew()</code>.</p> <p>Ignore the <code>IllegalStateException</code> if the session has been invalidated by other threads.</p>
CR134414	<p>When a <code>Serializable</code> Servlet request attribute was added, and then overwritten it with a non-<code>Serializable</code> value, the original value masks the new one.</p> <p>A code change was made to try to remove the value from a <code>HashMap</code> table of serializable attributes if necessary, when replacing a <code>Serializable</code> value with a non-<code>Serializable</code> one.</p>
CR132522	<p>On a Web server without a default Web application, an HTTP request for a missing resource received a response that included an incorrect date header:</p> <p>HTTP/1.1 404 Not Found Date: Thu, 01 Jan 1970 00:00:00 GMT</p> <p>This header is not valid according to section 14.18 of RFC2616. A code change resolved the problem.</p>

---

CR Number	Description
CR173042, CR110910	<p>HttpClusterServlet threw this exception, when KeepAliveEnabled was set to true after a large file download was canceled.</p> <p>Analysis revealed that when a client canceled a file download, the remaining data was left in the inputStream. If the socket was recycled for a subsequent request, the servlet read the remaining data, resulting in the exception. The problem was solved with a code fix to drain the inputStream and if the download is canceled we will read this remaining data.</p>
CR133558	<p>ServletContext.setAttribute threw a NullPointerException when a null value was passed to it.</p> <p>The problem was resolved by a code change that calls removeAttribute() when a null value is passed to the setAttribute().</p>
CR129211	<p>Using ServletOutputStream.write(byte) to write more than the buffer size caused infinite loop.</p> <p>A code change resolved the problem by updating the check for boundary conditions when the buffer is full and autoflush is set to false.</p>
CR128986	<p>After a protocol exception, the server hung while trying to process a new license because of a problem with the ensureContentLength method.</p> <p>A code change ensured that the server no longer hangs during this process.</p>
CR128420	<p>breakUpAndWriteItOutAsNecessary() tried to separate a manifest header to enforce a maximum of 72 bytes per line, and wrote one line to an outputStream at a time. The cause of this problem was that the start offset for the new line is wrong.</p> <p>A code change resolved the problem.</p>
CR127836	<p>A JSP in a subcontext of the root context was precompiled when deployed as a Web application packaged in a WAR file, but not if it was deployed in exploded format. JSPs now precompile when deployed in an exploded format.</p>
CR127090	<p>Jsp used several MBeans with attributes that had the same names, thus when they were displayed, it looked as though there were duplicates, when in fact, there were none.</p> <p>The display labeling of the MBean attributes was changed for the mbeans that had the same attribute names. They are now distinguishable.</p>
CR116209	<p>When a Serializable ServletContext attribute was added and then overwritten with a non-Serializable value, the original value masked the new value.</p> <p>When replacing a Serializable value with a non-Serializable one, WebLogic Server now removes the original value from the attributes hashtable.</p>

---

## 6 *Resolved Problems*

---

<b>CR Number</b>	<b>Description</b>
CR121343	<p>A race condition arose during the computation of a secondary JVMID when more than one frame was used. It appeared that the computation of the secondary JVMID was resetting the member variable value by one thread, causing the race condition. Following a code change, the computation of the secondary JVMID no longer leads to the race condition.</p>
CR120932	<p>The HTML page served by the CertificateServlet displayed the wrong value for the radio button—it displayed 2048 for the radio button, but using the button resulted in a 1024 bit certificate.</p> <p>The radio button now produces the correct 2048 bit certificate.</p>
CR091831	<p>The WebLogic Server implementation of HttpURLConnection did not check whether keep-alive connection had been timed out on the server side when using POST method, resulting in the error: Connection aborted by peer: socket write error on flush.</p> <p>Checks were added to ensure that the HttpClient is non-null before updating the timestamp.</p>
CR108034	<p>Inappropriate error messages generated by a user breaking a connection have been suppressed.</p>

## JVM

CR Number	Description
CR132228	<p>Harmless IOExceptions were not being suppressed on Japanese locales.</p> <p>WebLogic Server now sets the locale to C by default when enabling nativeIO. Customers who do not want to set the locale to C can use the system property -Dweblogic.nativeIO.useDefaultLocale=true</p> <p>On non-english locales, the harmless exceptions are now suppressed.</p>

## Operations, Administration and Management

CR Number	Description
CR122811	<p>A change to support piping of passwords into weblogic.Admin. That change locked up the System.in stream, preventing automated scripts from working correctly.</p> <p>The change that locked the System.in. stream has been corrected and scripts can now be automated.</p>
CR097343	<p>When a deployment was done where the targets were across two different platforms (such as unix and windows), the deployment files were not found.</p> <p>The deployment location was not being localized for the local machine type. If one deployed a webapp from an administration server running on windows and targeted it to a unix based machine, the location of the deployment was malformed for that platform.</p> <p>The deployment path has now been localized for each machine type.</p>
CR190237	<p>The WebLogic Server MBeanHome method getAllMBeans() threw an exception when running in a WebLogic Portal environment.</p> <p>WebLogic Server MBeanHome Helper needed additional knowledge of WebLogic Portal mbean class locations and a correction to existing logic.</p> <p>The ClassNotFoundException exception is no longer thrown in the WebLogic Portal environment.</p>
CR190212	<p>when an exception was thrown while retrieving attributes via MBeanServer.getAttributes(), any exceptions thrown by the JMXServer were handled and a null value was returned.</p> <p>WebLogic Server now rethrows the exception instead of returning a null value.</p>

<b>CR Number</b>	<b>Description</b>
CR188030	<p>CounterMonitor.stop() was synchronized causing a deadlock.</p> <p>Synchronization was removed, as called for in the documentation, eliminating the deadlocks.</p>
CR187170	<p>When the attributes of an array type changed(meaning certain elements of the array were added or deleted or both), AttributeAdd/Remove notifications were not being sent out. Thus certain deployments were not being carried out properly.</p> <p>The WebLogic Server now sends AttributeAdd/Remove notifications, in addition to the AttributeChange notification, when an array type attribute values change.</p>
CR113085	<p>When the value of SqlStmntProfilingEnabled was set through MBeans or manually in config.xml file and then you tried to retrieve that value through MBeans it did not show the value. Instead, it returned a message, "It appears that no attributes have been specified for this MBean".</p> <p>Exposing the SqlStmntProfilingEnabled attribute of JDBCConnectionPoolMBean eliminated the error.</p>
CR104435	<p>The KernelDebugMBean options, DebugDGCEnrollment and ForceGCEachDGCPeiod, could not be enabled. There was no way to "set" the enable attribute in weblogic.management.configuration.KernelDebugMBean.</p> <p>The setter signature did not take boolean for these attributes, so the values for these flags could not be changed from the default.</p> <p>These attribute setters now accept boolean.</p>
CR081349	<p>When printing the ExecuteQueueRuntimeMBean, the ExecuteThreads attribute was not being displayed correctly with the weblogic.Admin tool.</p> <p>The entries in the ExecuteThreads attribute are now displayed as a readable string value.</p>
CR094219	<p>When Domain and Server logfile names were date and time based (SimpleDateFormat), the following features were not fully functional:</p> <ol style="list-style-type: none"><li>1. These type of logs were not being rotated.</li><li>2. The total number of log files were not restricted even when the FileCount was set. There was no code in place to delete the older log files because the FileFilter could not list these files because of their file name patterns.</li></ol> <p>File rotation has been implemented for these types of file names. After each rotation, WLS now checks to see whether the files exceed the limit, if restricted. It then deletes the older log files.</p>

## Plug-Ins

CR Number	Description
CR180724	<p>The initial cookie was created through web server one and sent to cluster one. When it hit the application again it went through web server two and instead of being directed to cluster 1 it went to cluster 2 and created a new session.</p> <p>The WLCrossOverEnabled functionality now works correctly in WebLogic Server.</p>
CR132699	<p>The Apache 2.0.x plugin for Redhat Linux AS 2.1 returned an incorrect status code. The problem was resolved with a code change, which sets the status code to 500 when the backend WebLogic Server instance is not available.</p>
CR178792	<p>Apache) HTTP requests can contain either one of the following headers: Content-Length or Transfer-Encoding</p> <p>Requests with a Transfer-Encoding header set to "chunked" were failing with an IO error.</p> <p>Code was added to support requests using the Transfer-Encoding header set to "chunked".</p>
CR190562	<p>Requests were not retried when the plug-in encountered a broken pipe error on Solaris while sending post data to WebLogic Server.</p> <p>WebLogic Server now throws a HALF_OPEN_SOCKET_RETRY exception when sendPostData reports a broken pipe on Solaris</p>
CR189251	<p>Under load, Segmentation errors occurred while retrieving plugin Properties for a virtual host.</p> <p>Replacing the strtok API with strchr, as the strtok API is not thread safe, eliminated the errors.</p>
CR182971	<p>ServerList was deleted after every DNSRefreshInterval which resulted in a core dump.</p> <p>WebLogic Server now does a dns lookup of all the servers in the list and updates the ServerInfo structure if any server has changed from the last time it was checked.</p>
CR187282	<p>Because the plugin did not follow a part of the HTTP1.1 specification, which states that if a request/response contains both a Content-Length header as well as a Transfer-Encoding: Chunked header, the Content-Length header MUST be ignored, there was a unique scenario involving a recycled connection from the pool that sometimes caused an error.</p> <p>WebLogic Server now returns contentLength as -1 if CTE is on.</p>

## 6 Resolved Problems

---

CR Number	Description
CR095984	<p>When the file size exceeded 30KB, a DNS error message of IE 6.0 was received instead of a 413 message when using the NSAPI plug-in.</p> <p>The behavior of MaxPostSize configuration is now the same with or without a plug-in.</p>
CR188811 CR188808	<p>WLExcludePathOrMimeType did not work correctly when a request had a query string.</p> <p>Requests such as <code>http://webserver:port/weblogic/something.jsp?value=123</code> were not excluded and requests such as <code>http://webserver:port/weblogic/something.do?name=test.jsp</code> were not forwarded.</p> <p>The plug-in now ignores query strings while checking for excludes.</p>
CR187578	<p>When KeepAliveEnabled ON was configured in <code>httpd.conf</code>, <code>KeepAliveSecs</code> defaulted to 20. This default setting could not be changed.</p> <p>Code was added to ensure <code>KeepAliveSecs</code> is configurable.</p>
CR187577	<p>When using multiple <code>Location</code> tags in a <code>VirtualHost</code> tag, the Apache plug-in generated a strange URL if the requests matched two more <code>Locations</code>.</p> <p>The Apache plug-in no longer uses regular expression match, unless specified.</p>
CR186470	<p>When using the IIS plug-in, the creation of a large number of new connections through a firewall resulted in an HTTP status 302, and the connection was closed.</p> <p>WebLogic Server now recycles the connections if the HTTP status code is 302.</p>
CR186148	<p>When the Apache plug-in encountered a missing page, it was returning a 500 error, rather than the correct 503 error.</p> <p>The plug-in now returns the correct error.</p>
CR185668	<p>When using the Apache plug-in to proxy to multiple clusters using <code>MatchExpressions</code>, the <code>PathTrim</code> attribute was failing to trim off the segment of the url used to direct the request.</p> <p>Reimplementing <code>MatchExpressions</code> parsing without using the <code>strtok</code> API corrected the problem.</p>
CR185089	<p>The IIS plug-in was sending an Http status code of 500 (internal Server Error) when it encountered a <code>WRITE_ERROR_TO_CLIENT</code> exception due to a connection closed by the client.</p> <p>The IIS plug-in no longer sends Http status code of 500 when a <code>WRITE_ERROR_TO_CLIENT</code> exception is caught.</p>

CR Number	Description
CR180417	<p>If a cookie was part of the POST data then plugin would corrupt the post data while extracting the cookie.</p> <p>Code was added to fix the cookie extraction from Post Data.</p>
CR136816	<p>If the PRIMARY server could not be located, then the request was served by the next available server in the list. It ignored the Secondary server.</p> <p>If the Primary server can not be located, but the Secondary server is present then the request will be forwarded to Secondary server rather than being served by another server on the list.</p>
CR183390	<p>WebLogic Server was throwing an exception from inside the catch block which sometimes caused iPlanet to fail.</p> <p>WebLogic Server no longer throws an exception from the catch block.</p>
CR183311	<p>When Apache was stopped while using a single-thread multi-process module, it would try to stop the timer thread first. This timer thread never existed, thus a core dump occurred.</p> <p>WebLogic Server no longer creates timer threads when Apache is being used with a single-thread multi-process module.</p>
CR183188	<p>The ISAPI plug-in was unable to handle requests with the Transfer-Encoding header set to chunked.</p> <p>Functionality was added to enable ISAPI to handle such requests.</p>
CR182434	<p>Headings passed to rq-&gt;srvhdrs were not entirely in lower-case instead of mixed case.</p> <p>Content-type, content-length and transfer-encoding headers are now passed to NSAPI entirely in lower case.</p>
CR175787	<p>WebLogic Server was throwing CONNECTION_REFUSED errors with the Iplanet plugin.</p> <p>After polling a socket for WRITE operation, if the state of the socket is in any one of the following states, then the plug-in will not throw aCONNECTION_REFUSED exception.</p> <p>Valid states are:</p> <p>POLLOUT</p> <p>POLLWRBAND</p> <p>POLLWRNORM</p>

<b>CR Number</b>	<b>Description</b>
CR173985	<p>(Apache) The plug-in was dropping sessions after WebLogic Server was restarted. This is because Apache 1.3.x creates multiple child processes to handle incoming requests. Each process maintains its own serverlist where each server entry is uniquely identified by the JVMID (provided by WebLogic Server, and which is updated when a request is successfully processed). Whenever a server instance is restarted, it generates a new JVMID. So a request whose cookie contains a new JVMID will fail to locate the corresponding primary server plug-in if the JVMID is not refreshed in the plug-in.</p> <p>A code fix ensures that if the JVMIDs extracted from cookie do not match with the ones stored in the serverlist, then WebLogic Server will try to refresh the JVMIDs once again.</p>
CR136968	<p>Weblogic Server was not accepting more than one header when the response.addHeader method was used.</p> <p>The plug-in now allows WWW-Authenticate to have multiple values.</p>
CR135002	<p>In an Apache configuration with multiple virtual hosts, if only one of the virtual hosts was configured with SecureProxy=ON for the WebLogic Server plug-in, and the other virtual hosts did not use SecureProxy or WLProxySSL, the virtual hosts with no SSL configured saw that the plug-in attempted an SSL connection with the backend WebLogic Server. This caused a performance problem.</p> <p>A new argument was added to an internal method to determine if a SSL connection needs to be initiated.</p>
CR134413	<p>Apache plugin caused a duplicated http header and body for the 302 response. There was no problem between the plugin and backend servers, but the Apache server added an additional 302 response.</p> <p>Code was added which reverted the return value of the request_handler method to OK.</p>
CR132840	<p>Apache access logs improperly recorded a 200 code rather than a 500 error when application servers were down. A code change resolved the problem.</p>
CR132426	<p>When the plug-in parameter "QueryFromRequest" was used in the httpd.conf file it gave the following syntax error during Apache Server startup</p> <p>Syntax error on line 971 of C:/Program Files/Apache Group/Apache2/conf/httpd.conf:Invalid command 'QueryFromRequest', perhaps misspelled or defined by a module not included in the server configuration.</p> <p>The Apache 2.0 plug-in now supports the "QueryFromRequest" parameter.</p>

---

<b>CR Number</b>	<b>Description</b>
CR131640	WebLogic Server was requested to provide a plug-in for the Sun One 6.1 web server. The SUN ONE 6.1 Web Server is now fully supported.
CR180236	<p>The release 8.1 SP02 plug-in with client certificates reported the following error: Failed to parse the client certificate in header: WL-Proxy-Client-Cert. Ignoring this certificate. java.security.cert.CertificateException: Could not parse certificate: java.io.EOFException.</p> <p>The error occurred because the plug-in truncated the WL-Proxy-Client-Cert header when sending it to the WebLogic Server instance.</p> <p>The problem was resolved with a code fix.</p>
CR179537	<p>(ISAPI) The IIS proxy plug-in caused heap corruption on the Microsoft Windows platform.</p> <p>The problem was resolved with an internal code fix.</p>
CR177707	<p>When using the release 7.0 SP02 plug-in with client certificates, WebLogic Server worked correctly. However, after an upgrade to release 8.1 SP02, the server log reported the following error:</p> <p>Failed to parse the client certificate in header: WL-Proxy-Client-Cert. Ignoring this certificate. java.security.cert.CertificateException: Could not parse certificate: java.io.EOFException</p> <p>The error occurred because the 8.1 SP02 plug-in truncated the WL-Proxy-Client-Cert header when it sent it to the server instance.</p> <p>The code was changed so that WL-Proxy-Client-Cert is lazily added to the request sent to WebLogic Server.</p>
CR175989	<p>(Apache) The Apache server generated core dumps when using the worker (multi-threaded) option instead of the prefork (only multi-process) option.</p> <p>This was resolved by fixing the Locking and Unlocking logic.</p>
CR175672	<p>(Apache) The Apache server was hanging when the WebLogic plug-in tried to open the wlproxy log file, even though Debug was OFF.</p> <p>The code has been fixed so that the log file is not set if debugging is turned off.</p>
CR174777	<p>[iPlanet] POST_TIMEOUT errors occurred in the iPlanet log file due to a broken pipe.</p> <p>Code was added to throw a HALF_OPEN_SOCKET_RETRY exception if an EPIPE error is encountered while sending POST data to WebLogic Server.</p>
CR174431	(NSAPI) The Iplanet plug-in now gracefully handles an EINTR OS error.

---

## 6 Resolved Problems

---

CR Number	Description
CR173653	<p>When the <code>WLExcludePathOrMimeType</code> property was defined within a <code>Location</code> tag, it should not have had a global scope. (However, when the property was defined outside a <code>Location</code> tag, then it should have had a global scope.)</p> <p>This was resolved by a code change to ensure that the <code>WLExcludePathOrMimeType</code> property is applied only to the requests that match the appropriate <code>Location</code> path for the defined property.</p>
CR173581 CR173878	<p>(Apache) The plug-in was logging a confusing "error page is unavailable" log message to the <code>apache.error.log</code>, even when the client had closed the connection.</p> <p>This was resolved by a code change that commented out the erroneous log message.</p>
CR172497	<p>(NSAPI) The "<code>pluginparams.ErrorPageTest</code>" failed when attempting a proxy by extension.</p> <p>The workaround for the <code>ErrorPage</code> to be loaded locally is to use the <code>WLExcludePathOrMimeType</code> property if proxying by MIME type.</p> <p>For the SUN One Web Server, version 6.1, some additional configuration steps are required:</p> <ol style="list-style-type: none"><li>1. Remove <code>NameTrans fn="ntrans-j2ee" name="j2ee"</code> from the default object in the <code>obj.conf</code> file.</li><li>2. Remove <code>Init fn="load-modules" shlib="C:/Sun/WebServer6.1/bin/https/bin/j2eeplugin.dll" shlib_flags="(global now)"</code> from the <code>magnus.conf</code> file.</li></ol> <p>For more information, refer to Sun's documentation, at <a href="http://docs.sun.com/source/817-1831-10/agjava.html#wp1084_323">http://docs.sun.com/source/817-1831-10/agjava.html#wp1084_323</a></p>
CR172072	<p>Provided an enhancement to the <code>WLExcludePathOrMimeType</code> parameter allowing it to be used at the <code>Location</code> tag level.</p> <p><code>WLExcludePathOrMimeType</code> parameter can now be defined locally at the <code>Location</code> tag level as well as globally. When the property is defined locally, it does not override the global property but defines a union of the two parameters.</p>
CR171978	<p>When the <code>FilterPriorityLevel</code> was set in the <code>iisforward.ini</code> file, the forwarding path was broken.</p> <p>A code fix was implemented to ensure that when a virtual host was not defined in the <code>iisforward.ini</code> file, the <code>iisproxy.ini</code> file from the same location as where the <code>iisforward.dll</code> file was loaded is used.</p>

<b>CR Number</b>	<b>Description</b>
CR133641	<p>iPlanet users experienced a problem with host name verification and received the following:</p> <p>INFO: Host () doesn't match (), validation failed</p> <p>ERROR: SSLWrite failed</p> <p>A code fix resolved this issue.</p>
CR130060	<p>A performance problem in the IIS plug-in has been resolved in Service Pack 5 by a code change that causes the plug-in to check whether data equivalent to a specified content length has already been read.</p>
CR129471	<p>In previous service packs, the Apache plug-in did not recognize the WLTempDir parameter. This has been corrected.</p>
CR129342	<p>The ISAPI plug-in sent the WL-PATH-TRIM HTTP Header value to a WebLogic Server in place of the WL-PATH-TRIM value.</p> <p>WebLogic Server now receives the correct value.</p>
CR129138	<p>When the NSAPI plug-in performed name resolution on backend WebLogic Server instances, name resolution used sysGetHostByName, which called getHostByName, which called internal methods that had maximum limits for open file descriptors, causing name resolution sometimes to fail.</p> <p>A fix to cookie parsing and the substitution of JVMIDs to locate primary and secondary servers resolved the problem.</p>
CR129026, CR129323	<p>A memory leak in the ISAPI plug-in was fixed by a code change.</p>
CR127973	<p>The ISAPI plug-in sometimes failed after adding a persistent cookie to a servlet session.</p> <p>A correction to the cookie parsing code resolved the problem.</p>
CR127658	<p>[ISAPI] When a connection was retrieved from the pool, but the WebLogic Server had already closed the connection, then the HALF_OPEN_SOCKET_RETRY exception was raised.</p> <p>Requests will be now be retried after receiving the HALF_OPEN_SOCKET_RETRY exception.</p>

## 6 *Resolved Problems*

---

<b>CR Number</b>	<b>Description</b>
CR084303	<p>WebLogic Server proxy plug-ins restrict the HTTP commands that can be submitted from the client to the server. The validation rules in the plug-in code now allow the following HTTP commands that are needed for WebDAV implementations:</p> <p>DELETE GET HEAD OPTIONS POST PUT *COPY LOCK MKCOL MOVE PROPFIND PROPPATCH SEARCH UNLOCK</p>
CR127231	<p>A request did not fail over to the next available server in the cluster after receiving 503 HTTP status. The same server was tried repeatedly until a <code>READ_ERROR_FROM_SERVER</code> or a <code>CONNECTION_REFUSED</code> exception was raised.</p> <p>Code now marks the server as failed on getting a 503 HTTP status error, gets the next available server and re-sends the request. All requests now successfully fail over to next available server.</p>

---

## RMI-IIOP

CR Number	Description
CR188748 CR180749	<p>During the repository id generation of getter and setter operations, sometimes an extra '_' was added if the getter or setter contained a reserved keyword. This sometimes resulted in the failure of a remote call.</p> <p>WebLogic Server now ensures it truncates extra underscores.</p>
CR175112	<p>The method used to send IIOP messages was unsynchronized leading to corruption of the underlying data if multiple threads tried to send and receive data at the same time.</p> <p>The method was made synchronized and the product now works correctly in multi-threaded environments.</p>

## RMI

CR Number	Description
CR177353	<p>When a application client code cached the remote stub and invoked a remote method on a SLSB deployed to a cluster, the behavior was that each call refreshed the list which tracked the cluster nodes where the remote object is available. This list was used to failover the calls if any of the node failed with a recoverable exception. The issue here was that failover did not work when the entire cluster was restarted while the application client had cached the stub from a previous invocation.</p> <p>The retry logic in the failover algorithm was incorrect. This logic originally allowed n-1 number of retries in a cluster with n nodes. When the entire cluster restarted, the cached stub would have stale list. And the retry logic scanned though the stale list and exhausted all the retry attempts. In the last attempt it would have potentially refreshed the list. Even though the client side stub now had a new copy of the list it did not attempt to failover as it has already reached n-1 attempts limit.</p> <p>The remote stub cached in the application client now ensures it refreshes the list only when remote method invocation fails on all the nodes in the existing list. Then stub is given one last chance for failover if the list got refreshed. If this last chance does not succeed the stub will throw an exception to the application otherwise failover will continue to work as advertised transparently.</p>

# Samples

---

<b>CR Number</b>	<b>Description</b>
CR183199	Building the examples on Windows XP resulted in a <code>StringIndexOutOfBoundsException</code> . Upgrading to Ant 1.6.1 eliminated the build problem.
CR183527	In WebLogic Server 6.1 SP06, the <code>examples.ejb 1.1</code> package, <code>examples/ejb/building.html</code> provides incorrect instructions for building samples, and incorrectly describes how the samples build script works. The instructions and description refer to previous version of the build script which used <code>build.cmd</code> or <code>build.sh</code> . The current version of the build script ( <code>build.xml</code> ) uses ant. To run <code>build.xml</code> , type <code>ant</code> in the directory for the sample you wish to build. To understand how the build script works, refer to the comments in <code>build.xml</code>
CR132509	When using the .NET C# example included with the MedRec application ( <code>%WEBLOGIC_HOME%\samples\server\medrec\src\clients\CSharpClient\bin\Release\CSharpClient.exe</code> ), the client successfully retrieves a patient record, but when attempting to "Save Changes" from the client application, a date field error was flagged. A code change fixed the date validation.

---

# Security

---

<b>CR Number</b>	<b>Description</b>
CR045135	Some internal WebLogic Server objects that are bound into the JNDI tree were insufficiently protected. Some of these objects control key functions of the server. Simply unbinding the appropriate objects renders the server unreachable. It was possible to leave the server operational but to have some confidential data transmitted to a foreign object rather than the internal WebLogic Server object. The patches available in this release control access to the JNDI tree. See <a href="#">Security Advisory BEA04-65.00</a> .

---

CR Number	Description
CR103309	<p>Whenever an HTTP or HTTPS request was made of the server, information about the WebLogic Server version was supplied.</p> <p>The default setting for returning a Web Server version upon HTTP or HTTPS request has been changed from true to false.</p> <p>See <a href="#">Security Advisory BEA04-70.00</a>.</p>
CR125520	<p>A deadlock occurred between two threads when using Netscape LDAP classes - LDAPConnection and LDAPConnThread. This deadlock grew until all the default execution threads were deadlocked with LDAPConnThreads.</p> <p>After updating to the new Netscape SDK, deadlocks no longer occur.</p>
CR128940	<p>A protection problem occurred when WebLogic Server was running on an operating system that required case-sensitive filenames and cross-mount directories containing Web applications from an operating system that did not support case-sensitive filenames. For example, this problem occurred when mounting Windows-based Web applications onto a Linux-based WebLogic Server.</p> <p>This resulted in some URL patterns in the web.xml file being incorrectly evaluated so that access control was compromised. The associated patch adds configuration options to specify that Web application files should be treated as case-insensitive even though the operating system may be running in case-sensitive mode.</p> <p>See <a href="#">Security Advisory BEA04-67.00</a>.</p>
CR123860	<p>The constructor method for a flat group used by a custom realm expected a case-sensitive value. When a case-sensitive value was not found, a NullPointerException was thrown.</p> <p>WebLogic Server no longer throws a NullPointerException when the value for the CachingRealm MBean value is null.</p>
CR136544	<p>A set of enhancements have been added to the WebLogic Server command-line utilities and Administrative ant tasks to eliminate the need for a system administrator to enter a clear-text password. With these fixes, the WebLogic Server command-line utilities and Administrative ant tasks now create and read user-managed configuration files. Like the boot.properties files, this new capability relies on strong encryption and file system protections for security.</p> <p>See <a href="#">Security Advisory BEA04-68.00</a>.</p>
CR179663 CR188110	<p>A NullPointerException occurred when using the WebLogic Server Administration Console to list LDAP users.</p> <p>Code was added to eliminate the NullPointerException.</p>

## 6 Resolved Problems

---

CR Number	Description
CR174299	<p>The Use Java attribute on the SSL tab in the WebLogic Server Administration Console was not working properly.</p> <p>The behavior of the attribute has been fixed.</p>
CR172187	<p>A password echo occurred intermittently on the Linux operating system when booting WebLogic Server using the Administrative Console.</p> <p>No-echo capabilities have been added to resolve the issue. See the README file in the patch for further documentation.</p> <p>See <a href="#">Security Advisory BEA04-69.00</a>.</p>
CR172567	<p>If the certificate name was the same for both an Administration Server and remote Managed Server, WebLogic Server used the certificate for the Administration Server for the remote Managed Server.</p> <p>The SSL Listen thread has been updated so that it only loads local certificates.</p>
CR161836	<p>WebLogic Server was setting the AuthenticatedUser in the session rather than the AuthenticatedUserName.</p> <p>Servlet authentication has been changed so that it sets the AuthenticatedUsername resolved in the ClassCast exception.</p>
CR127722	<p>If host name verification failed but the setExpectedName() method was defined, WebLogic Server ignored the host name defined in the method and dropped the connection.</p> <p>WebLogic Server now performs the host name verification check first. If that check fails, WebLogic Server uses the ExpectedName() method to verify the hostname. If that verification fails, the connection is rejected.</p>
CR112619	<p>A NullPointerException was being thrown when an Applet attempted to obtain an initial context using the t3s protocol.</p> <p>The basic constraints check for applets was updated, and this eliminated the NullPointerException.</p>
CR124746, CR175051, CR175045	<p>Following a code change, WebLogic Server does NOT support HTTP TRACE requests by default. If you want to turn on HTTP TRACE support, in WebLogic Server 6.1 and above, set HttpTraceSupportEnabled="true" in cluster/server. In 5.1, set weblogic.htpd.httpTraceSupport.enabled=true.</p> <p>See <a href="#">Security Advisory BEA04-48.01</a>.</p>

<b>CR Number</b>	<b>Description</b>
CR112220	WebLogic Server was verifying the hostname of the administration server against the administration certificate in the nodemanager, thus causing an error. Hostname verification in the nodemanager now only checks against the nodemanager hosts file.

---

### SNMP

---

<b>CR Number</b>	<b>Description</b>
CR113122	The value that the WebLogic Server SNMP agent returned for sysUpTime did not accurately report the duration since the SNMP agent had been initialized. The duration since initialization is now accurately reported.
CR103678, CR109869	When SNMP information was collected using a third-party collector task, the following message was logged: <Error> <SNMP Agent> <000000> < Unable to set Entry Field Value...> A code fix was implemented to resolve this issue.

---

### WLEC

---

<b>CR Number</b>	<b>Description</b>
CR121105	WLEC clients were experiencing the following problems with WLEC connection pools: Corruption of the ConnectionPool display in the administration console COMM_FAILURES caused by unregistered endpoints A code fix was implemented to resolve these issues and improve the overall performance of WLEC connection pools.

---

### WebLogic Tuxedo Connector

---

<b>CR Number</b>	<b>Description</b>
CR133654	Fixed a problem with a race condition between the tBridge spawning threads for each redirect definition and the WTC Service setting an initialized flag to true.
CR185588	Fixed a problem in which the last character in a double byte character set is being cut in the middle and turning into 'null' on the WLS side after passing through WTC.

---

# WebLogic Server 6.1 Service Pack 6 Solutions

This section lists problems resolved in WebLogic Server 6.1 SP06.

- Classloader
- Cluster
- Connector
- Console
- Core
- Deployment
- EJB
- JDBC
- jDriver
- JMS
- JNDI
- JSP
- JTA
- Node Manager
- OA&M (Operations, Administration, and Management)
- Plug-ins
- RMI
- RMI/IIOP
- Security
- Servlets

## **6** *Resolved Problems*

---

- SNMP
- Web Services
- WTC-ATMI
- XML

## Classloader

Change Request Number	Description
CR101547	<p>A memory leak occurred when deploying an Enterprise Application that contained a singleton class with static data. The problem did not occur when the singleton class was deployed as a Web Application.</p> <p>This problem was solved with a fix to the application classloader.</p>
CR104513	<p>With the fix for CR069506 introduced in WebLogic Server 6.1 Service Pack 2, the classloader failed to find files when one manifest finder recursively referenced other manifest entries. The problem was solved by changing <code>ClassLoader.getResources()</code> to handle this case.</p>
CR111924	<p>WebLogic Server threw a <code>ClassNotFoundException</code> when processing a user-defined exception thrown from an EJB on a remote server to an EJB on the local server. This occurred even though the user-defined exception was included in the classloader for the local EJB. The problem occurred because the socket reader that processed the exception used the system classloader, rather than the application classloader.</p> <p>This problem was solved with a code fix.</p>
CR124348	<p>Client programs could not use <code>java.lang.reflect.Proxy</code> to access proxy objects deployed in WebLogic Server, unless the proxy classes were added to the system classpath. If an object did not reside in the system classpath, the client would receive a <code>ClassNotFoundException</code>.</p> <p>The <code>resolveProxyClass()</code> method was implemented to load interfaces from the application-specific classloader as well as the system classloader.</p>

## Cluster

---

<b>Change Request Number</b>	<b>Description</b>
CR103807	WebLogic Server 6.1 SP04 threw a null pointer exception at <code>weblogic.rmi.cluster.BasicReplicaList.add(BasicReplicaList.java:61)</code> . The problem occurred when a managed server was restarted after a failure. Upon restart, the NPE occurred and all server instances in the cluster had to be restarted. The error was related to a timing issue, and was solved with a code fix.
CR104430	Clustered servers could lose sessions when a client switched to a third server in the cluster from the first and second servers, which had been the client's primary and secondary servers. A process would remove the session from the first two servers, and when the client switched back to the primary server, the primary server looked for the session on the secondary server, instead of properly looking on the third server.  A code fix resolved the problem by causing the session to be recreated from session information on the third server, completely removing the session from the primary and secondary servers.

---

---

Change Request Number	Description
-----------------------	-------------

---

CR107471	<p>In WebLogic Server 6.1 SP02, when using HTTP Tunneling with a cluster, the client got this message:</p> <pre>java weblogic.Admin -url http://colma:17683 -username system -password password PING 10 &lt;May 29, 2003 10:14:18 AM PDT&gt; &lt;Error&gt; &lt;RJVM&gt; &lt;000515&gt; &lt;execute failed java.net.ProtocolException: Tunneling result not OK, result: 'DEAD', id: '0'java.net.ProtocolException: Tunneling result not OK, result: 'DEAD', id: '0'   at weblogic.rjvm.http.HTTPClientJVMConnection.receiveAndDispatc h(HTTPClientJVMConnection.java:422)   at weblogic.rjvm.http.HTTPClientJVMConnection.execute(HTTPClie ntJVMConnection.java:305)   at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213   at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)&gt; Failed to connect to &lt;urldeleted&gt; due to: &lt;urldeleted&gt;:Bootstrap to &lt;urldeleted&gt; failed. It is likely that the remote side declared peer gone on this JVM] The problem occurred because the plug-in tried to round-robin each tunnel request to the next server. The request did not stick to the same server. The problem was solved with a code fix to ensure state was maintained in the client by setting the jsessionid cookie and sending it back to the plugin.</pre>
----------	---

---

<b>Change Request Number</b>	<b>Description</b>
CR108127	<p>In WebLogic Server 6.1 SP04, an error occurred when trying to update a secondary session. Given three clustered server instances: Server A, Server B, and Server C.</p> <ol style="list-style-type: none"><li>1) Hit Server A, PRIMARY A, SECONDARY C</li><li>2) Hit Server C, PRIMARY C, SECONDARY B</li><li>3) Hit Server A again</li></ol> <p>This error occurred on Server A:</p> <pre>&lt;Jun 4, 2003 4:24:56 PM PDT&gt; &lt;Debug&gt; &lt;Cluster&gt; &lt;Error updating secondary for 3535724191309537720 on 7312270160516507840S:&lt;IPaddressdeleted&gt;: [7005,7005,7002,7002,7005,7002,-1]:&lt;IPaddressdeleted&gt;:7005, &lt;IPaddressdeleted&gt;:7005,&lt;IPaddressdeleted&gt;:7005:mydomain:mserver3. Re-creating secondary.&gt;</pre> <p>This error occurred on Server C:</p> <pre>&lt;Jun 4, 2003 4:24:56 PM PDT&gt; &lt;Info&gt; &lt;Cluster&gt; &lt;Lost 0 replication updates of object 3535724191309537720. Re-fetching secondary.&gt;</pre> <p>The problem was solved by a code change.</p>
CR112874	<p>In WebLogic Server 6.1 SP03, when a client of a stateful session bean accessed a bean deployed on a cluster configured for weight-based load-balancing, and the Managed Servers with the highest and second highest weight are killed in that order, the client gives the following message</p> <pre>&lt;Jul 22, 2003 1:32:56 AM IST&gt; &lt;Warning&gt; &lt;Kernel&gt; &lt;No replica in list has the expected weight. Reverting to round-robin&gt;</pre> <p>When the Managed Servers are restarted, the load balancing algorithm was switched to round-robin.</p> <p>Analysis revealed that the replica list was getting updated when a Managed Server went down, but due to a race condition the max weight in RichReplicaList was not reset properly.</p> <p>A code change to recompute max weight whenever the replica list size changes solved the problem.</p>

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR116954	<p>The web application container returned 404 messages for InitJVMID requests from the HTTPClusterServlet, resulting in log files filling up with messages:</p> <pre>####&lt;Sep 6, 2002 4:56:43 PM EDT&gt; &lt;Debug&gt; &lt;HTTP&gt; &lt;petunia&gt; &lt;msslpetunia&gt; &lt;ExecuteThread: '13' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;101147&gt; &lt;HttpServer(887891,null default ctx,msslpetunia) found no context for "/WLDummyInitJVMIDs" .</pre> <p>The problem was solved by sending InitJVMID requests to the internal web application which is always deployed on WebLogic Server.</p>
----------	--

---

## Connector

---

<b>Change Request Number</b>	<b>Description</b>
CR109463	<p>When a transaction obtained connections to two resource adapters and both adapters used the same resource manager, WebLogic Server cleaned up only one connection. The connections were closed before the committing the transaction. This problem appeared as a <code>javax.resource.spi.ResourceAllocationException</code>.</p> <p>The problem occurred because the transaction manager ignored the second connection that used the same resource manager. This was solved with a code fix.</p>
CR121418	<p>When allocating a new connection, the connector container used the descriptor MBean to obtain configuration information. Because the descriptor MBean resides on the Administration Server, Managed Servers required the Administration Server to be available in order to allocate the new connection. If the Administration Server was down, the process of allocating new connections would throw an exception:</p> <pre>weblogic.rmi.extensions.RemoteRuntimeException</pre> <p>The container was modified so that it stores configuration information locally after an adapter is deployed. Managed Servers use this local configuration information, rather than the descriptor MBean, for allocating new connections.</p>
CR125555	<p>When WebLogic Server Service Pack 5 obtained MetaData information from a Resource Adapter, it did not check the return value for nulls before using the object. This could cause a <code>NullPointerException</code> and connection failure.</p> <p>The code was modified to check the return value for null after calling <code>getMetaData()</code>. If an adapter's implementation of <code>ManagedConnection.getMetaData()</code> returns a null value, WebLogic Server no longer throws a <code>NullPointerException</code> and no MetaData is logged.</p>

---

---

## Console

---

Change Request Number	Description
CR100006	<p>WebLogic Server sometimes displayed the following <code>NullPointerException</code> when a domain contained two Administration Servers and you tried to access the servers via the Administration Console:</p> <pre>java.lang.NullPointerException at weblogic.management.console.helpers.UrlHelper.buildIconList( UrlHelper.java:149) at weblogic.management.console.helpers.UrlHelper.getIcon(UrlHel per.java:174) at weblogic.management.console.tags.SmartNavNodeTag.getIcon(Sma rtNavNodeTag.jacva:111) at weblogic.management.console.tags.SmartNavNodeTag.inferStuffF romContext(SmartNavNodeTag.java:96) at weblogic.management.console.tags.SmartNavNodeTag.doStartTag( SmartNavNodeTag.java:44) at weblogic.management.console.webapp.__domain.__nav.__jspService (__nav.java:301) at weblogic.servlet.jsp.JspBase.service(JspBase.java:27) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:262) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:198) at weblogic.servlet.internal.RequestDispatcherImpl.forward(Requ estDispatcherImpl.java:250)  [...]</pre> <p>This problem was solved with a code fix.</p>

---

<b>Change Request Number</b>	<b>Description</b>
CR102824	<p>After upgrading from WebLogic Server SP02 to SP04 on Solaris 8, when customer logs into the console they get the following exception in the log files. The console continues to work.</p> <pre>java.lang.NullPointerException at weblogic.management.console.utils.ConsoleComparator.compare( ConsoleComparator.java:81) at java.util.Arrays.mergeSort(Arrays.java:1176) at java.util.Arrays.sort(Arrays.java:1123) at java.util.Collections.sort(Collections.java:116) at weblogic.management.console.utils.MBeans.sort(MBeans.java:11 03) at weblogic.management.console.tags.DeclareBeanSetTag.doStartTa g(DeclareBeanSetTag.java:99) at weblogic.management.console.webapp.__nav_services.__j spService(__nav_services.java:982) at weblogic.servlet.jsp.JspBase.service(JspBase.java:27) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:262) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:198) at weblogic.servlet.internal.RequestDispatcherImpl.include(Requ estDispatcherImpl.java:490) at weblogic.servlet.internal.RequestDispatcherImpl.include(Requ estDispatcherImpl.java:316) at weblogic.servlet.jsp.PageContextImpl.include(PageContextImpl .java:116) at...</pre> <p>Analysis revealed that MBeans were returning a null <code>displayName</code>. The problem was solved with a code fix.</p>
CR105598	<p>New testing with an Active Directory LDAP uncovered a problem where a call to <code>group.isMember()</code> would fail if a member name contained a comma character (,). This problem was solved with a code fix.</p>

---

## Core

---

<b>Change Request Number</b>	<b>Description</b>
CR071415	<p>In WebLogic Server 6.1 SP02, when the WebLogic Server security manager starts and sets the policy file, the path <code>/usr/lib</code> was prepended to <code>javac</code> during JSP compilation, causing a <code>java.security.AccessControlException</code>.</p> <p>The problem was solved with a code fix.</p>
CR094410	<p>When <code>DebugHttp</code> was activated in <code>ServerDebugMBean</code>, WebLogic Server reported <code>SocketResetExceptions</code> as errors, rather than simple debug messages. To address this problem, <code>logMuxableSocketResetException()</code> was added to the message catalog for reporting this debug message.</p>
CR096091	<p>In WebLogic Server 6.1 SP04 and SP05, when starting WebLogic Server as a Windows service using a classpath from a file, the error</p> <pre>"Thread created successfully! Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/Server"</pre> <p>is thrown if classpath file is over about 2k length.</p> <p>The problem was solved by correcting an error related to reading the classpath from a file.</p>

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR102058	<p>When using URLClassLoader on the client, a remote method call resulted in a <code>NoClassDefFoundError</code>.</p> <pre>c:\java\java131_07\bin\java -classpath ".;client.jar;C:\home\ravia\weblogic\dev\src700\3rdparty\web logicaux.jar" URLTest qa146 9901 ejb20-statefulSession- TraderHome installadministrator installadministrator java.lang.NoClassDefFoundError: weblogic/rmi/extensions/server/Stub at java.lang.ClassLoader.defineClass0(Native Method) at java.lang.ClassLoader.defineClass(ClassLoader.java:488) at java.security.SecureClassLoader.defineClass(SecureClassLoade r.java:106) at weblogic.utils.classloaders.GenericClassLoader.findLocalClas s(GenericClassLoader.java:401) at weblogic.utils.classloaders.GenericClassLoader.findClass(Gen ericClassLoader.java:162) at java.lang.ClassLoader.loadClass(ClassLoader.java:294) at java.lang.ClassLoader.loadClass(ClassLoader.java:250) at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:310 ) at java.lang.Class.forName0(Native Method) at java.lang.Class.forName(Class.java:190) at weblogic.utils.classfile.utils.CodeGenerator.generateClass(C odeGenerator.java:71) at ...</pre> <p>The problem was solved by setting <code>ThreadContextClassLoader</code> as the parent for <code>AugmentableSystemClassLoader</code>.</p>
----------	---

---

CR103525	<p>In WebLogic Server 6.1 SP04, this error occurred:</p> <pre>weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[ Old socket closed and released FD before FDRecord still exists,...</pre> <p>The problem was solved with a code change to Ensure that socket closes outside the muxer are handled gracefully.</p>
----------	---

---

---

Change Request Number	Description
-----------------------	-------------

---

CR103774	<p>The Solaris and Windows 2000 versions of WebLogic Server logged socket exceptions similar to:</p> <pre>&lt;Apr 3, 2003 11:40:07 AM EST&gt; &lt;Error&gt; &lt;socket&gt; &lt;000424&gt; &lt;IOException on socket: weblogic.servlet.internal.MuxableSocketHTTP@1eca988 - idle timeout: '30000' ms, socket timeout: '0' ms, fd: 53 java.net.SocketException: Connection reset java.net.SocketException: Connection reset     at java.net.SocketInputStream.read(SocketInputStream.java:168)     at weblogic.socket.PosixSocketMuxer.readBytesProblem(PosixSocketMuxer.java:876)     at weblogic.socket.PosixSocketMuxer.deliverGoodNews(PosixSocketMuxer.java:767)     at weblogic.socket.PosixSocketMuxer.processSockets(PosixSocketMuxer.java:694)     at weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:23)     at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213) ) ad.run(ExecuteThread.java:189) &gt;</pre> <p>However, these messages do not indicate a defect with the server or application. The code changed so that the above exceptions are displayed only when the server is in debug mode.</p>
----------	---

---

CR103967	<p>In WebLogic Server 6.1 SP04, child threads of WebLogic Server execute threads did not inherit the correct context classloader. This problem common occurred when a servlet or an ejb created a timer (<code>java.util.Timer</code>).</p> <p>Analysis revealed that WebLogic Server execute threads maintained their own context classloader and child threads of these execute threads did not inherit the context because <code>java.lang.Thread</code> directly assigned the member variable of its own to the child threads.</p> <p>The problem was solved with a code fix.</p>
----------	---

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR104218	<p>In WebLogic Server 6.1 SP04, the following exception occurred on a Solaris 8 server with patch for CR100665_61sp4.jar:</p> <pre>&lt;Apr 23, 2003 3:36:27 PM CDT&gt; &lt;Error&gt; &lt;Posix Performance Pack&gt; &lt;Uncaught Throwable in processSockets java.util.ConcurrentModificationException at java.util.HashMap\$HashIterator.nextEntry(HashMap.java:750) at java.util.HashMap\$EntryIterator.next(HashMap.java:792) at java.util.HashMap.putAllForCreate(HashMap.java:408) at java.util.HashMap.clone(HashMap.java:624) at java.util.HashSet.clone(HashSet.java:217) at weblogic.socket.PosixSocketMuxer.closeSockets(PosixSocketMuxer.java:486) at weblogic.socket.PosixSocketMuxer.processSockets(PosixSocketMuxer.java:589) at weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:24) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)</pre> <p>The problem was solved with a code fix.</p>
----------	--

---

---

Change Request Number	Description
-----------------------	-------------

---

CR105426	<p>WebLogic Server ships with a file named <code>wlntio_g.dll</code>, which is a debug version of <code>wlntio.dll</code> (and is required only for debugging purposes). WebLogic Server 6.0 Service Pack 2 included the wrong version of <code>wlntio_g.dll</code>, which could cause stack traces similar to the following when used for debugging:</p> <pre>&lt;NT Performance Pack&gt; NATIVE.malloc(): allocated at 0x08DAD008 numAllocs 1 numFrees 0 NATIVE: start io on 1972 with addr 0x08dad008 NATIVE: GetQueuedCompletionStatus on 1972 with addr 0x08dad008 &lt;May 6, 2003 11:17:19 AM EDT&gt; &lt;Error&gt; &lt;NT Performance Pack&gt; &lt;failure in processSockets() - GetData: 'weblogic.socket.NTSocketMuxer\$GetData - native pointer: '0', numBytes: '0'' java.lang.NoSuchFieldError: fd     at weblogic.socket.NTSocketMuxer.getNextSocket(Native Method)         at weblogic.socket.NTSocketMuxer.processSockets(NTSocketMuxer.j ava:589)         at weblogic.socket.SocketReaderRequest.execute(SocketReaderRequ est.java:24)         at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139 )         at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)&gt;</pre> <p>Service Pack 6 now includes the correct version of <code>wlntio_g.dll</code>.</p>
----------	---

---

---

**Change Request Number**    **Description**

---

CR105516	<p>In WebLogic Server 6.1 SP04, stateful session EJB failover did not work when multiple failovers were required</p> <p>In a three node cluster a JSP creates or calls a replica-aware stateful session EJB. The remote EJB stub is stored in the http session. After each call to the stateful session EJB the updated EJB stub is also updated in the http session to reflect any changes for the EJB replica list (primary/secondary).</p> <p>The following call sequence with the same browser window leads to a <code>java.rmi.ConnectException</code> when only one node of the cluster survives:</p> <ol style="list-style-type: none"><li>1. All three cluster nodes are running.</li><li>2. Making a call to node1, create EJB and store remote in http session (http session replication is enabled)</li><li>3. Kill node1</li><li>4. Make a call to the secondary node2, the EJB remote is retrieved from the replicated http session and the call to the EJB works fine. After this EJB call again the remote is stored in the http session.</li><li>5. Kill node2</li><li>6. Make a call to node3, get the EJB remote from http session.</li></ol> <p>WebLogic Server tries to lookup the EJB on node2 and does not try to use node3 (this should now be the new secondary?). The following exception is thrown on node3:</p> <pre>java.rmi.ConnectException: Could not establish a connection with -3088833905169218734S:172.23.64.38:[7001,7001,7002,7002,7001,7002,-1]:mydomain:managed2, java.rmi.ConnectException: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused: connect; No available router to destination     at weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java:275)     at weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java:408)     at weblogic.rmi.internal.BasicRemoteRef.getOutboundRequest(BasicRemoteRef.java:97)     at weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwareRemoteRef.java:255)</pre> <p>The problem was solved with a code fix.</p>
----------	---

---

<b>Change Request Number</b>	<b>Description</b>
CR105814	<p>In WebLogic Server 6.1 SP04, a customer saw <code>close_waits</code> running under AIX. Analysis revealed that when WebLogic Serve 6.0 clients connected to a WebLogic Server 6.1 instance, WebLogic Server leaked sockets. This occurred when WebLogic Server attempted to close the <code>MuxableSocket</code> via the muxer without registering the socket with the muxer—the socket was rejected after being claimed by <code>t3</code>, before it could be registered with the muxer. The problem was corrected with a code fix.</p>
CR106957	<p>In WebLogic Server 6.1 SP04, running with IBM's MQWorkflow on AIX 5.2., invocation of an MQWorkflow Java API by a servlet in WebLogic Server resulted in an error in MQWorkflow. The problem was not exhibited under Window, if Native IO (Performance Pack) was disabled, or if the <code>libmuxer.so</code> library was removed from the classpath.</p> <p>Analysis revealed that WebLogic Server did not set the “language code”, an encoding parameter, to “en-us” as it should, but to “c”. The problem was corrected with a code fix.</p>

<b>Change Request Number</b>	<b>Description</b>
CR107598	<p>In WebLogic Server 6.1 SP03 and SP04, restart of the Administration Server resulted in this error:</p> <pre>java.rmi.ConnectException: This RJVM has already been shutdown The problem was reproduced consistently when beforeing these steps:</pre> <ol style="list-style-type: none"><li>1. Configure a cluster with server instances on at least two machines.</li><li>2. Deploy a sample EJB to cluster.</li><li>3. Restart the Administration Server.</li><li>4. Untarget the EJB from cluster</li><li>5. Retarget the EJB to cluster.</li></ol> <p>The following error resulted:</p> <pre>&lt;Feb 5, 2003 7:03:27 PM PST&gt; &lt;Info&gt; &lt;J2EE&gt; &lt;Undeployed: ejb_basic_statelessSession&gt; java.rmi.ConnectException: This RJVM has already been shutdown -7152222714009328 37S:172.17.25.250:[7001,7001,7002,7002,7001,7002,-1]:mydomain:myserver at weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java:280) at weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java:408) at weblogic.rmi.internal.BasicRemoteRef.getOutboundRequest(BasicRemoteRef.java:97) at weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.java:125) at weblogic.rmi.internal.ProxyStub.invoke(ProxyStub.java:35) at \$Proxy7.getMBeanServer(Unknown Source) at weblogic.management.internal.MBeanProxy.getAttribute(MBeanProxy.java:253) The problem was solved with a code change to remove stale MBean references.</pre>
CR109339	<p>In WebLogic Server 6.1 SP05, a security exception initializing kernel in applets. A code fix solved the problem.</p>

---

---

Change Request Number	Description
-----------------------	-------------

---

CR109590	<p>In WebLogic Server 6.1 SP04, running with AIX, a method call made over IIOP to a remote object hung.</p> <p>The customer has a server (non-WebLogic Server) hosting remote objects. An EJB running on WebLogic Server calls a remote object on the non-WebLogic Server over IIOP. The method call hung indefinitely. The method call was simple—it returns a boolean type. The thread dump showed:</p> <pre>"ExecuteThread: '11' for queue: 'default'" (TID:0x300C12A8, sys_thread_t:0x35649A58, state:CW, native ID:0x1011) prio=5 at java.lang.Object.wait(Native Method) at java.lang.Object.wait(Object.java:429) at weblogic.iiop.SequencedRequestMessage.waitForData(SequencedR equestMessage.java:24) at weblogic.iiop.EndPointImpl.sendReceive(EndPointImpl.java:641 ) at weblogic.iiop.OutboundRequestImpl.sendReceive(OutboundReques tImpl.java:43) at weblogic.iiop.IIOPRemoteRef.invokeInternal(IIOPRemoteRef.jav a:128) at weblogic.iiop.IIOPRemoteRef.invoke(IIOPRemoteRef.java:90) at weblogic.rmi.internal.ProxyStub.invoke(ProxyStub.java:35) at \$Proxy70.isRunning(Unknown Source) at java.lang.reflect.Method.invoke(Native Method) at weblogic.iiop.IIOPInvocationHandlerImpl.invoke(IIOPInvocatio nHandlerImpl.java:285) at \$Proxy71.isRunning(Unknown Source) at...</pre> <p>Analysis revealed a classloader problem. A code fix resolved the problem.</p>
----------	---

---

Change Request Number	Description
CR110347	<p>In WebLogic Server 6.1 SP05 a lookup on context (and assigning to object) throws <code>IllegalArgumentException</code> if EJB stubs is not in classpath</p> <p>The problem occurred with a stateful session EJB (examples.ejb20.basic.statefulSession, shipped with WebLogic Server) deployed to a single WebLogic Server server. A JDNI lookup was performed on the EJB home object and the returned value was assigned to the class object. The client classpath contained <code>weblogic.jar</code> and no client classes.</p> <p>This code was used to perform the lookup on the context.</p> <pre>Context ctx = new InitialContext(ht); Object objref = ctx.lookup("ejb20-statefulSession-TraderHome");</pre> <p>The lookup succeeded for latter succeeds for WebLogic Server 6.1 SP03 but failed on WebLogic Server 6.1 SP05. The exception was:</p> <pre>java.lang.IllegalArgumentException: java.lang.IllegalArgumentException: interface examples.ejb20.basic.statefulSession.</pre> <p><code>TraderHome</code> was not visible from the class loader With verbose classloading set for the JVM, it was found that the <code>examples.ejb20.basic.statefulSession.TraderHome</code> class was successfully loaded by the In WebLogic Server 6.1 SP03, although it was not in the classpath. With In WebLogic Server 6.1 SP05, the failure occurred when attempting to load that class.</p> <p>The problem was solved by a code change to ensure the <code>ClientRuntimeDescriptor</code> uses the current classloader, if it is a <code>GenericClassLoader</code>.</p>
CR110892	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-35.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-35.jsp</a>.</p>
CR116712	<p>In WebLogic Server 6.1 SP04, the <code>WLECCONNECTIONRuntime</code> MBeans were not updated when an <code>INVOKE</code> was issued on <code>WLECCONNECTIONPoolRuntime</code> using Admin tool.</p> <p>Analysis revealed that when <code>resetConnectionPool()</code> was invoked on the <code>WLECCONNECTIONPoolRuntime</code> MBean, the pool was not reset. Old connections were not removed</p> <p>The problem was solved with a code change. Now, the mbeans that correspond to old connections are unregistered before the <code>resetConnectionPool()</code> invocation.</p>

Change Request Number	Description
CR117200	<p>In WebLogic Server 6.1 SP05, a deadlock in <code>weblogic.socket.PosixSocketMuxer</code> was detected when a Managed Server could not connect to the Administration Server. This is an extract from the thread dump:</p> <pre> 1wasLKDEADLOCK      Deadlock detected!!! NULL NULL 2LKDEADLOCKTHR Thread "ExecuteThread: '0' for queue: '__weblogic_admin_rmi_queue'" (0x8461B8A0) 3LKDEADLOCKWTR is waiting for: 4LKDEADLOCKMON sys_mon_t:0x84D183A8 inflaming: 0x00000000: 4LKDEADLOCKOBJ weblogic.socket.PosixSocketMuxer\$FDRecord@31FD4B10/31FD4B18: 3LKDEADLOCKOWN which is owned by: 2LKDEADLOCKTHR Thread "ExecuteThread: '98' for queue: 'default'" (0x767D26E0) 3LKDEADLOCKWTR which is waiting for: 4LKDEADLOCKMON sys_mon_t:0x83AD8C18 inflaming: 0x00000000: 4LKDEADLOCKOBJ weblogic.rjvm.ConnectionManagerServer@31FA3430/31FA3438: 3LKDEADLOCKOWN which is owned by: 2LKDEADLOCKTHR Thread "ExecuteThread: '0' for queue: '__weblogic_admin_rmi_queue'" (0x8461B8A0) </pre> <p>The following exception is also thrown:</p> <pre> java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@2a7512ad - id: '8419466038107054512S:10.32.197.52:[7001,7001,-1,-1,7001,-1, -1]:bossapp:AdminServer' connect time: 'Fri. Aug 01 09:43:07 CST 2003'' has already been shut down at weblogic.rjvm.ConnectionManager.getOutputStream(ConnectionMa nager.java(Compiled Code)) at... </pre> <p>Analysis revealed that <code>PosixSocketMuxer</code> hit the deadlock due to contention for the <code>FDRecord</code> and <code>ConnectionManager</code> locks. One thread holds the <code>FDRecord</code> lock and waits for a <code>ConnectionManager</code> lock, which is held by another thread which is waiting for the <code>FDRecord</code> lock to do a cleanup.</p> <p>A code fix removed the contention. Now, the <code>FDRecord</code> lock is not held during dispatch.</p>

<b>Change Request Number</b>	<b>Description</b>
CR122280	<p>For context-wide sessions, WebLogic Server did not check if an object was Serializable before placing it in the session. This could lead to the error:</p> <pre>Could not deserialize context attribute     java.io.NotSerializableException: com.app.name     at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.j ava(Compiled Code))     at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.ja va(Compiled Code))     at weblogic.servlet.internal.AttributeWrapper.getObject(Attribu teWrapper.java(CompiledCode))     at weblogic.servlet.internal.WebAppServletContext.getAttribute( WebAppServletContext.java(Compiled Code))     at weblogic.servlet.internal.WebAppServletContext.getAttribute( WebAppServletContext.java(Compiled Code))</pre> <p>The code was modified so that <code>setAttribute()</code> checks for a serializable object before adding it to the session.</p>
CR123571	<p>When starting several T3 clients on the same machine simultaneously, there was a possibility that two or more of the clients could obtain the same JVMID, cause exceptions or hanging on the clients. The problem occurred only when starting multiple T3 clients on the same machine at the same time. This problem was solved by modifying the code used to generate JVMIDs.</p>
CR124763	<p>Prior to WebLogic Serve Service Pack 6, all server instances in a cluster used the server listen port as the multicast port for cluster communication. There was no way for a cluster to use a multicast port different from the server listen port.</p> <p>This Service Pack introduces a new, optional Java property, <code>-Dweblogic.cluster.multicastport=port_number</code>, to specify the multicast port used in a cluster. To use the new property, all cluster members must specify the same <code>-Dweblogic.cluster.multicastport</code> value in their startup scripts.</p> <p>If the <code>-Dweblogic.cluster.multicastport</code> is not specified at startup time, servers continue to use the listen port as the multicast port for cluster communication.</p>

# Deployment

---

<b>Change Request Number</b>	<b>Description</b>
CR102324	<p>Prior to this Service Pack release, WebLogic Server did not log a nested exception associated with a deployment error. The logged text indicated only the deployment error, as in:</p> <pre>&lt;Mar 28, 2003 1:10:51 AM PST&gt; &lt;Error&gt; &lt;J2EE&gt; &lt;Error deploying application application_name: Caught IOException for path=path_to_application&gt;</pre> <p>The code was modified to log the nested exception as well as the deployment error itself, as in:</p> <pre>&lt;Mar 28, 2003 1:10:51 AM PST&gt; &lt;Error&gt; &lt;J2EE&gt; &lt;Error deploying application_name: with deployment error Caught IOException for path=path_to_application and nested error java.io.FileNotFoundException: File not found&gt;</pre>

---

# EJB

---

Change Request Number	Description
CR056023	<p>A CMP2.0 EJB accessing an Oracle database did not timeout when the database was locked.</p> <p>The problem was exhibited when running the <code>ejb2.0 cmp</code> example, with <code>trans-timeout-seconds</code> to a non-zero value such as “10”, against a locked database table (the table was locked using SQLplus “lock table ejbaccounts in exclusive mode”). When the table was locked, and the client ran, the client hung indefinitely trying to create an account. After the database lock was released using SQLplus, a rollback exception occurred in the client. The rollback should occur when at the end of the transaction timeout.</p> <p>Analysis revealed that, when the timeout period expired, and the timer sent a <code>TransactionRolledBack</code> message to the database, the database did not release the lock.</p> <p>The problem was resolved, for the Oracle thin driver, for row-level locks by:</p> <ul style="list-style-type: none"><li>■ Doing a <code>cancelAllStatementsBeingUsed</code> in the <code>internalRollback</code> method of the <code>weblogic.jdbc.jts.Connection</code> class before calling a rollback on the connection.</li><li>■ Introducing a new method, <code>cancelAllStatementsBeingUsed</code> in the <code>weblogic.jdbc.common.internal.ConnectionEnv</code> class that iterates through all the open statements and calls cancel on them for the connection being rolled back, ignoring the SQL exception.</li></ul> <p>The fix works only for the Oracle thin driver, for row-level locks.</p>

---

Change Request Number	Description
CR098343	<p>In WebLogic Server 6.1 SP02 and SP04, JDBC connections were not returned to the pool when a transaction was distributed over two server instances. The JDBC attribute <code>KeepXAConnTillTxComplete</code> was set to <code>true</code>. This sequence of events led to the problem:</p> <ol style="list-style-type: none"> <li>1. Start a user transaction on one server instance 1.</li> <li>2. Put a JMS message into a persistent queue. The JMS server and queues are on server instance 2.</li> <li>3. Update an EJB on server instance 2.</li> <li>4. Commit the transaction.</li> </ol> <p>No error messages were generated. The updates were performed and the message was placed in the JMS queue. However, some connections remained 'in use' in the ejb connection pool. After repeating the sequence of steps several times, the connection pool runs out of connections.</p> <p>The problem did not occur if the actions are executed on a single server instance, or if the JMS message is placed in the queue outside the transaction scope.</p> <p>The problem was solved with a code change to release the connection in the <code>aftercompletion</code> callback.</p>
CR099041	<p>In WebLogic Server 6.1 SP04, a customer reported random generation of the following exception while invoking methods in a particular EJB in his application:</p> <pre>java.lang.ClassCastException: java.lang.String at weblogic.utils.classfile.DoubleKey.equals(DoubleKey.java:24) at java.util.Hashtable.get(Hashtable.java:318) at weblogic.utils.classfile.ConstantPool.find(ConstantPool.java :58) at weblogic.utils.classfile.ConstantPool.getUtf8(ConstantPool.j ava:251) at weblogic.utils.classfile.expr.ClassInfo.addField(ClassInfo.j ava:86) at weblogic.utils.classfile.expr.DotClassExpression.code(DotCla ssExpression.java:34) at weblogic.utils.classfile.expr.InvokeExpression.code(InvokeEx pression.java:31) at weblogic.utils.classfile.expr.ExpressionStatement.code(Expre ssionStatement.java:19) at weblogic.utils.classfile.expr.TryCatchStatement.code(TryCatc hStatement</pre> <p>The problem was corrected with a code fix to the <code>equals()</code> method in <code>weblogic.utils.classfile.DoubleKey</code>.</p>

## 6 *Resolved Problems*

---

<b>Change Request Number</b>	<b>Description</b>
CR099626	In WebLogic Server 6.1 SP03 ejbc generated invalid SQL for an <code>ejbSelect</code> query. The query failed because table aliases were generated incorrectly, resulting in this error: <code>ORA-00904: invalid column name</code> The problem was corrected with a code fix.
CR099760	A field optimization was implemented for EJB 1.1 CMP beans, so that fields that are updated, but whose value did not change, are not written to back to the database. The optimization is only done for primitives and immutable objects.

---

---

Change Request Number	Description
-----------------------	-------------

---

CR100246	<p>A <code>LockTimeoutException</code> occurred during a test, upon attempt to remove a stateful EJB. The sequence of operations was:</p> <ol style="list-style-type: none"><li>1. During test within the remote method of an entity bean, create a stateful session bean (local/localhome interfaces).</li><li>2. Call business methods on the stateful session bean, which in turn create different types of beans.</li><li>3. Try to remove the stateful session bean.</li></ol> <p>On calling remove the following exception is thrown:</p> <pre>Nov 18, 2002 12:51:33 AM PST&gt; &lt;Info&gt; &lt;EJB&gt; &lt;010049&gt; &lt;EJB Exception in method: remove: weblogic.ejb20.locks.LockTimedOutException: The lock request from EJB:LibraryInfoEJB with primary key:85335648642269185 timed-out after waiting 0 ms. The transaction or thread requesting the lock was:Thread[ExecuteThread: '6' for queue: 'default',5,Thread Group for Queue: 'default']. weblogic.ejb20.locks.LockTimedOutException: The lock request from EJB:LibraryInfoEJB with primary key:85335648642269185 timed-out after waiting 0 ms. The transaction or thread requesting the lock was:Thread[ExecuteThread: '6' for queue: 'default',5,Thread Group for Queue: 'default'].     at weblogic.ejb20.locks.ExclusiveLockManager\$LockBucket.lock(Ex clusiveLockManager.java:449)     at weblogic.ejb20.locks.ExclusiveLockManager.lock(ExclusiveLock Manager.java:259)     at weblogic.ejb20.manager.StatefulSessionManager.acquireLock(St atefulSessionManager.java:248)     at weblogic.ejb20.manager.StatefulSessionManager.acquireLock(St atefulSessionManager.java The problem was solved by adding a new <code>weblogic-ejb-jar.xml</code> deployment descriptor element, <code>&lt;allow-remove-during-transaction&gt;</code>. When this element is set to true, the exception does not occur</pre>
----------	--

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR101918	The following error was thrown during ejb20 home methods tests, on NT with hotspot, and Native IO disabled:
----------	---

```
java.lang.IllegalStateException: zip file closed at
java.util.zip.ZipFile.ensureOpen(ZipFile.java:377)atjava.util.zip.ZipFile.getEntry(ZipFile.java:138)at
eblogic.utils.classloaders.ClasspathClassFinder.getSourcesInternal(ClasspathClassFinder.java:225)at
weblogic.utils.classloaders.ClasspathClassFinder.getSource(ClasspathClassFinder.java:155)at
weblogic.utils.classloaders.MultiClassFinder.getSource(MultiClassFinder.java:53)at
weblogic.utils.classloaders.MultiClassFinder.getClassSource(MultiClassFinder.java:45)at
weblogic.utils.classloaders.GenericClassLoader.findLocalClass(GenericClassLoader.java:303)at
weblogic.utils.classloaders.GenericClassLoader.findClass(GenericClassLoader.java:161)weblogic.rmi.internal.BasicRuntimeDescriptor.getClientRuntimeDescriptor(BasicRuntimeDescriptor.java:526)...
```

Analysis revealed that the problem was related to uncanceled triggers during undeployment. The following correction solved the problem:

- `rmic` code was fixed to remove the static classloader entry in the `Utilities` class at the end of `ejbc`
  - `ExecuteThread` was fixed to clean the contextclassloader before re-using the thread.
  - `ejb` code was fixed for canceling the triggers when the `cmp/bmp` with exclusive concurrency was undeployed.
-

Change Request Number	Description
CR102028	<p>DB QueryString was generated incorrectly when using the Oracle thin driver for Connection Pool creation.</p> <ol style="list-style-type: none"> <li>1. Create a connectionPool using oracle thin driver.</li> <li>2. Run the test case ejb20/reactions/FindersTest/testStringFunctionLOCATE().</li> <li>3. The above test case can be run from the ejb20/reactions/finder.ts file.</li> <li>4. Make sure the EJBs “ComputerEJB”, “EmployeeEJB” and “FinderEmployeeEJB” use this connection Pool, created using thin driver.</li> <li>5. When this is run, we get “Invalid Character” error.</li> </ol> <p>The stack trace is</p> <pre> javaw.exe: java.lang.ClassNotFoundException: javax.ejb.FinderException: Problem in findByNameEquals while preparing or executing statement: 'weblogic.jdbc.jts.PreparedStatement@55c5eb': java.sql.SQLException: ORA-00911: invalid character java.sql.SQLException: ORA-00911: invalid character at oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java: 134) at oracle.jdbc.ttc7.TTIOer.processError(TTIOer.java:289) at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:573) at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol...</pre> <p>The problem was corrected with a code fix to generate the correct SQL query</p>
CR102180	<p>In WebLogic Server SP03, ejbc failed while compiling stubs and skeletons for a .jar with 100 EJBs. The following exception occurred:</p> <pre> java.io.IOException: CreateProcess: c:\VisualCafe\bin\sj.exe -classpath c:\java\java130\jre\lib\rt.jar; c:\java\java130\jre\lib\i18n.jar;c:\java\java130\jre\lib\sun rsasign.jar;c:\java\java130\jre\classes; c:/java/java130/lib/tools.jar;D:/weblogic/dev/src/3rdparty/j connect/jConnect.jar;D:/weblogic/src_130sj/license;D:/weblog ic/dev/src/3rdparty/oracle/816/classes12.zip; D:/weblogic/src_130sj/classes;D:/weblogic/src_130sj/lib/xmlx .jar;D:/weblogic/src_130sj/lib/ejb20.jar;D:/weblogic /dev/src/3rdparty/weblogicaux.jar;D:/weblogic/dev/src/3rdpar ty/cloudscape/lib/cloudscape.jar;D:/weblogic/src_130sj/class es_ifmx4; D:/weblogic/src_130sj/classes_mssql4;D:/weblogic/src_130sj/t ools;c:/java/java130/jre/lib/rt.jar;D:\raviWork\ejb...</pre> <p>The problem was a result of a compiler command line length limitation. It was solved by using the javac @tempfile feature, which allows file names to be passed to the compiler using a temporary file.</p>

<b>Change Request Number</b>	<b>Description</b>
CR102308	<p>In WebLogic Server 6.1 SP04, the Administration Console reported incorrect values for waiters for entity beans.</p> <p>The problem was solved by adding a <code>waiterCurrentCount</code> attribute that is incremented when a client starts waiting for a lock and decremented when the lock is acquired or the client times out.</p>
CR103047	<p>It was reported in WebLogic Server 6.1 SP03 that transaction timeouts on MDBs were not logged. When the time an MDB takes to process a message from a JMS Destination exceeds the transaction timeout limit, the transaction is rolled-back and the message is place backed on the destination for re-delivery, but no transaction-timeout or transaction-rollback messages were logged.</p> <p>This problem was resolved by a code change to the MDListener code to report the transaction timeouts.</p>

---

---

Change Request Number	Description
CR103978	<p>In WebLogic Server 6.1 SP02, a web application failed to obtain a remote object from a session under the following circumstances:</p> <ul style="list-style-type: none"><li>■ A web application deployed on the Administration Server of Domain 1.</li><li>■ An EJB was deployed to one Managed Server in a cluster in Domain 2.</li><li>■ The web application did a lookup on the EJB using the URL of the Managed Server to which the EJB was deployed.</li><li>■ The web application created a remote object and put it on the HttpSession</li><li>■ The web application failed to obtain the remote object from the session with the following exception:</li></ul> <pre data-bbox="327 597 1166 961">&lt;Apr 21, 2003 11:42:47 AM PDT&gt; &lt;Error&gt; &lt;HTTP Session&gt; &lt;Error reconstructing the EJBObject put into session for name: trader java.rmi.NoSuchObjectException: Unable to locate EJBHome: 'statelessSession.TraderHome' on server: 't3://acaoclust:7001 at weblogic.ejb20.internal.HomeHandleImpl.getEJBHome(HomeHandleImpl.java :80) at weblogic.ejb20.internal.HandleImpl.getEJBObject(HandleImpl.java:179) at weblogic.servlet.internal.session.SessionData.getAttribute(SessionData.java:390) at jsp_servlet.__remoteejb.__jspService(__remoteejb.java:119) at weblogic.servlet.jsp.JspBase.service(JspBase.java:27) at</pre> <p>The error occurred because the web application used the cluster URL to get the remote EJB object from the HttpSession, although EJB was deployed to a single Managed Server, not to the cluster.</p> <p>A code fix solved the problem. Now the EJB Handle (HomeHandle) uses the URL of the server to which the EJB is deployed, if the EJB home is not clusterable, as specified in the EJB's <code>home-is-clusterable</code> deployment element in the <code>weblogic-ejb-jar.xml</code> file. The cluster URL is used only when <code>home-is-clusterable</code> is true.</p>

---

---

Change Request Number	Description
-----------------------	-------------

---

CR104414	<p>In WebLogic Server 6.1 SP04, server-side ejbc compilation errors resulted when the classpath was long. This error resulted:</p> <pre>[EJBCompiler] : Compiling EJB sources Warning: UNIXProcess.forkAndExec native error: The parameter or environment lists are too long. &lt;Apr 24, 2003 2:55:27 PM GMT&gt; &lt;Error&gt; &lt;J2EE&gt; &lt;Error deploying application applicationname: Unable to deploy EJB: monitor/EMAServerManager.jar from monitor/EMAServerManager.jar: Compiler failed executable.exec(java.lang.String[javac, -nowarn, -classpath, .....long classpath which is not shown here..... ....)     at weblogic.ejb20.ejbc.EJBCompiler.compileEJB(EJBCompiler.java( Compiled Code))     at weblogic.ejb20.deployer.Deployer.runEJBC(Deployer.java(Inlin ed Compiled Code))     at weblogic.ejb20.deployer.Deployer.compileEJB(Deployer.java(Co mpiled Code))</pre> <p>The problem was solved by using in-line compilation by specifying the <code>-compilerclass</code> option, and using <code>noexit</code> instead of the <code>callcompile</code> flag so that the <code>compile</code> method of the <code>compilerclass</code> gets called.</p>
----------	---

---

---

Change Request Number	Description
-----------------------	-------------

---

CR105609	<p>A Null Pointer Exception was thrown when a WebLogic Server 6.1 SP04 servlet or stateless session bean invoked a stateful session bean deployed on WebLogic Server 6.1 SP02. The stack trace is:</p> <pre>java.lang.NullPointerException at weblogic.ejb20.internal.HandleImpl.readExternal(HandleImpl.java:101) at java.io.ObjectInputStream.inputObject(ObjectInputStream.java:1212) at java.io.ObjectInputStream.readObject(ObjectInputStream.java:386) at java.io.ObjectInputStream.inputClassFields(ObjectInputStream.java:2263) at java.io.ObjectInputStream.defaultReadObject(ObjectInputStream.java:519) at java.io.ObjectInputStream.inputObject(ObjectInputStream.java:1412) at java.io.ObjectInputStream.readObject(ObjectInputStream.java:386) at java.io.ObjectInputStream.readObject(ObjectInputStream.java:236) at java.util.ArrayList.readObject(ArrayList.java:531) at java.lang.reflect.Method.invoke(Native Method) at java.io.ObjectInputStream.invokeObjectReader(ObjectInputStream.java:2214) at java.io.ObjectInputStream.inputObject(ObjectInputStream.java:1411) at java.io.ObjectInputStream.readObject(ObjectInputStream.java:386) at java.io.ObjectInputStream.readObject(ObjectInputStream.java:236) at</pre> <p>Analysis revealed that the wire format of the handle is extended after WebLogic Server 6.1 SP02 to resolve handle issues related to differences for stateful session bean handles when the stub is stamped with the handle. In the case of stateless session beans, stubs are not added with handles.</p> <p>The problem was resolved by a code change to allow for the “no handle” case for stateless session beans.</p>
----------	---

---

Change Request Number	Description
CR106136	<p data-bbox="400 289 1233 344">In WebLogic Server 6.1 SP04, getter methods for a EJB 1.1 CMP bean did not call <code>isModified()</code> when <code>delay-updates-until-end-of-tx</code> was set to <code>false</code></p> <p data-bbox="400 357 1233 526">A session EJB called an entity EJB's getter methods. Both EJBs had container managed transaction with transaction attribute set to <code>Required</code>. Each call to a getter method is followed by a call to <code>ejbStore()</code> and <code>delay-updates-until-end-of-tx</code> was <code>false</code>. However, before calling <code>ejbStore</code> on the bean the container did not call the <code>isModified</code> method. The <code>isModified()</code> method was only called when the transaction committed.</p> <p data-bbox="400 539 1233 591"><code>ejbStore</code> should be called from <code>postInvoke()</code> depending on the result of the <code>isModified</code> method in the bean. The problem was solved with a code fix.</p>
CR111551	<p data-bbox="400 623 1233 678">In WebLogic Server 6.1 SP05, EJBC generated CMP code that caused a <code>java.lang.ClassCastException: oracle.sql.BLOB</code> error.</p> <p data-bbox="400 691 1233 883">Analysis revealed that the CMP RDBMS code generated for the query was incorrect. Prior to WebLogic Server 6.1 SP05, server-side JDBC went through the RMI drive, to the pool driver, then on to the DBMS driver. Starting in WebLogic Server 6.1 SP05, the RMI driver is now only invoked in external clients. Server-side client code accesses the pool driver directly, which returns the DBMS driver BLOBs directly, not wrapped in an RMI wrapper. This change requires that code cast directly to <code>oracle.sql.BLOB</code> instead of <code>weblogic.jdbc.common.OracleBlob</code>.</p> <p data-bbox="400 896 1233 948">The problem was solved by correcting the EJBC to generate code that reflects the SP05 and later behavior.</p>

Change Request Number	Description
CR111771	<p>In WebLogic Server 6.1 SP02, the database was left in inconsistent state when a bean is undeployed. The problem occurred in this invocation scenario:</p> <pre>webApp &lt;--&gt; OuterBean &lt;--&gt; InnerBean</pre> <p>The beans are stateless session beans, using container-managed transactions, with <code>trans-attribute=Required</code>. <code>OuterBean.m()</code> inserts a row in a table, then calls <code>InnerBean.m()</code>, which inserts a row in a different table. Since the scenario happens within the context of a single transaction, both tables have the same number of rows. The beans are packaged in different jars. <code>OuterBean</code> caches the home of <code>InnerBean</code> and everything is deployed on a single server. <code>InnerBean</code> is untargeted from the server during the course of the test. In this case, the tables are left in an inconsistent state: <code>OuterBean</code> no longer inserts any rows in its table while <code>InnerBean</code> still does.</p> <p>Analysis revealed that when <code>InnerBean</code> was undeployed, <code>TxManager.undeploy</code> rolls back in-flight transactions and marks the instance as dead, so that no further transactions can be enlisted. <code>OuterBean</code> and <code>InnerBean</code> were packaged in the same enterprise application, and calls were by reference and did go through RMI. The home and the bean are cached in <code>OuterBean</code>, so even after <code>InnerBean</code> has been undeployed, the remote methods called are serviced. After <code>InnerBean</code> was undeployed, when <code>OuterBean</code> called <code>InnerBean.insert()</code>, there was a transaction associated with the call—the one started by <code>OuterBean</code>. In the <code>preInvoke</code>, the <code>TxListener</code> is set up and the transaction is register with the <code>TxManager</code>. While registering with the <code>TxManager</code>, the instance was determined to be dead, and WebLogic Server rolled back the transaction and silently return. As no transaction is enlisted, the database connection obtained within the <code>InnerBean.insert()</code> is not within the called transaction, so the rollback done in <code>OuterBean</code> has no effect on the row inserted by <code>InnerBean</code>—resulting in the situation where <code>OuterBean</code> no longer inserts any rows in its table while <code>InnerBean</code> still does. Silently returning, after rolling back the transaction because the bean is undeployed is the cause of the problem.</p> <p>The problem was solved with a code change to check the deployment status in the <code>preInvoke</code> of the <code>BaseEJBManager</code>, thus preventing calls from reaching a bean that has been undeployed.</p>
CR115026	<p>In WebLogic Server 6.1 SP02, using MDBs listening on MQSeries, the MQSeries java threads on the WebLogic Server side were not correctly closed upon on an MQSeries failure and re-start of MQSeries. With every failure/restart the number of MQSeries AsyncThreads doubled.</p> <p>The problem was solved with improvements to the failure and cleanup code in <code>JMSConnectionPoller.createJMSConnection()</code>.</p>

## JDBC

<b>Change Request Number</b>	<b>Description</b>
CR098343	<p>JDBC connections were not returned to the pool when a transaction was distributed over two server instances. The JDBC attribute <code>KeepXAConnTillTxComplete</code> was set to true. This sequence of events led to the problem:</p> <ol style="list-style-type: none"><li>1. Start a user transaction on one server instance 1.</li><li>2. Put a JMS message into a persistent queue. The JMS server and queues are on server instance 2.</li><li>3. Update an EJB on server instance 2.</li><li>4. Commit the transaction.</li></ol> <p>No error messages were generated. The updates were performed and the message was placed in the JMS queue. However, some connections remained 'in use' in the EJB connection pool. After repeating the sequence of steps several times, the connection pool ran out of connections.</p> <p>The problem did not occur if the actions were executed on a single server instance, or if the JMS message was placed in the queue outside the transaction scope.</p> <p>The problem was solved with a code change to release the connection in the <code>aftercompletion</code> callback.</p>
CR099872	<p>The message for the Exception during commit of transaction stack trace exception contained the connection pool name, but not the data source name.</p> <pre>&lt;Exception during commit of transaction Xid=39:74a54046e2c2bb30(5962554),Status=Rolled back. [Reason=ja vax.transaction.xa.XAException: JDBC driver does not support XA, hence cannot be a participant in two-phase commit. To force this participation, set the EnableTwoPhaseCommit property on the corresponding JDBCDataSource property, to true. Pool = CatPhase1]...</pre> <p>The stack trace content was enhanced to include the name of the data source.</p>
CR103046	<p>When using a JDBC connection pool, issuing <code>Statement.close()</code> did not release all resources associated with the underlying <code>ResultSet</code>, which could lead to an <code>OutOfMemoryError</code>. This problem occurred only when closing the <code>Statement</code> via a connection pool. It did not occur when explicitly closing the <code>ResultSet</code> itself, or when closing a statement while directly using the driver.</p> <p>This problem was solved with a code fix.</p>

---

Change Request Number	Description
-----------------------	-------------

---

CR103299	<p>This Service Pack corrects an error in the leak detection code that caused the following stack trace:</p> <pre>[SerialConnection] : Connection Leak detected!!!!!!java.lang.Throwable: StackTrace at creation of connection: Start server side stack trace: java.lang.Throwable: StackTrace at creation of connection: at weblogic.jdbc.rmi.SerialConnection.&lt;init&gt;(SerialConnection.j ava:47) at weblogic.jdbc.pool.Connection.writeReplace(Connection.java:1 427) at java.lang.reflect.Method.invoke(Native Method)at java.io.ObjectStreamClass.invokeMethod(ObjectStreamClass.jav a:1615) at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.ja va:303) at weblogic.common.internal.ChunkedObjectOutputStream.writeObje ct(ChunkedObjectOutputStream.java:115) at weblogic.rjvm.MsgAbbrevOutputStream.writeObject(MsgAbbrevOut putStream.java:82) at weblogic.jdbc.common.internal.RmiDataSource_WLSkel.invoke(Un known Source) at weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.j ava:398) at weblogic.rmi.cluster.ReplicaAwareServerRef.invoke(ReplicaAwa reServerRef.java:114) at weblogic.rmi.internal.BasicServerRef\$1.run(BasicServerRef.ja va:339) at weblogic.security.service.SecurityServiceManager.runAs(Secur ityServiceManager.java:850) at weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServ erRef.java:334) at weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecu teRequest.java:30)at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:215 ) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:191) End server side stack trace</pre>
----------	--

---

<b>Change Request Number</b>	<b>Description</b>
CR103321	<p>In WebLogic Server 6.1 SP04, the <code>getConnectionsTotalCount()</code> method of <code>JDBCConnectionPoolRuntimeMBean</code> did not behave as expected. Instead of returning the total number of JDBC connections in the pool since instantiation, it returned the maximum number of connections since instantiation.</p> <p>The problem was resolved with a code fix the method.</p>
CR103421	<p>In WebLogic Server 6.1 Service Packs 3 and 4, transactions involving the <code>BigDecimal</code> datatype threw a <code>SQLWarning</code> exception if the transaction obtained a local <code>DataSource</code> before obtaining a remote <code>DataSource</code>. The text of the stacktrace was:</p> <pre>java.sql.SQLException: java.sql.SQLWarning: Exhausted Resultset Start server side stack trace: weblogic.common.internal.WLSQLWarning: Exhausted Resultset     at weblogic.jdbc.common.internal.DriverProxy.getWLSQLFromSQLException(DriverProxy.java:2     at weblogic.jdbc.common.internal.ResultSetProxy.execute(ResultSetProxy.java:716)     at weblogic.t3.srvr.ClientRequest.execute(ClientContext.java:769)     at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)     at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) End server side stack trace</pre> <p>This problem occurred because the T3 driver always queried the remote result set for <code>getBigDecimal()</code>, rather than querying the cached data.</p> <p>This problem was solved with a code fix.</p>
CR103619	<p>This Service Pack fixes a problem with the <code>isClosed()</code> method and JTS connections. The problems were detected while running Sun's CTS for JDBC.</p>
CR106767	<p>When applications try to close JDBC objects more than once, WebLogic Server now throws an exception.</p>

<b>Change Request Number</b>	<b>Description</b>
CR108580	<p>WebLogic Server failed to check for a null pool name before creating a ConnectionLeakProfile object, causing the exception:</p> <pre>java.lang.NullPointerException at java.lang.String.&lt;init&gt;(String.java:193) at weblogic.jdbc.common.internal.ConnectionLeakProfile.&lt;init&gt;(C onnectionLeakProfile.java:21) at weblogic.jdbc.rmi.internal.ConnectionImpl.unreferenced(Conne ctionImpl.java:144) at weblogic.rmi.internal.BasicServerRef\$UnreferencedExecuteRequ est.execute(BasicServerRef.java:702) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:234 at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:210)</pre> <p>This problem was solved with a code fix.</p>
CR112607	<p>If the URL or password for a connection pool was incorrect, WebLogic Server 6.1 Service Pack 4 still created a new JDBC connection pool with an initial capacity of 0. The code was modified to ensure that the creation of a JDBC connection pool fails if either the URL or password are incorrect.</p>
CR120455	<p>A memory leak was discovered in WebLogic Server 6.1 Service Pack 2. The leak occurred when using a TxDataSource to access a BLOB column on a database with the WebLogic XA driver. This problem was solved with a code fix.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR120531	<p>In WebLogic Server Service Pack 5, resetting a connection pool using <code>weblogic.Admin RESET_POOL</code> (or an equivalent API) caused an exception similar to the following if a connection was already released:</p> <pre>&lt;Aug 13, 2003 1:33:11 PM EST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;[WebAppServletContext(7096795, jdbc_webapp, /jdbc_webapp)] Servlet failed with Exception java.lang.Error: 1 Was already released:weblogic.jdbc.common.internal.Connection</pre> <p>After a garbage collection, the server would then display a connection leak warning:</p> <pre>&lt;Aug 13, 2003 1:33:19 PM EST&gt; &lt;Warning&gt; &lt;JDBC&gt; &lt;A JDBC pool connection leak was detected. A Connection leak occurs when a connection obtained from the pool was not closed explicitly by calling close() and then was disposed by the garbage collector and returned to the connection pool. The following stack trace at create shows where the leaked connection was created. Stack trace at connection create:     at weblogic.jdbc.pool.Connection.&lt;init&gt;(Connection.java:55)     at [...]</pre> <p>The code was fixed to eradicate the connection leak warning; the server still correctly displays the initial exception if a connection was already released at the time the pool is reset.</p>
----------	---

---

CR121671	<p>In WebLogic Server Service Pack 4, the <code>weblogic.jdbc.jta.ResultSet.getType()</code> method could recursively call itself and lead to a stack overflow exception:</p> <pre>java.lang.StackOverflowError at weblogic.jdbc.jta.ResultSet.checkIfClosed(ResultSet.java:106 ) at weblogic.jdbc.jta.ResultSet.getType(ResultSet.java:507) at weblogic.jdbc.jta.ResultSet.getType(ResultSet.java:508)</pre> <p>This problem was solved with a code fix.</p>
----------	---

---

<b>Change Request Number</b>	<b>Description</b>
CR125135	<p>By default, WebLogic jDriver for Oracle/XA Data Source set the value of the <code>oracleXATrace</code> parameter to true, rather than false. This caused the driver to create trace files of the form <code>xa_poolname*.trc</code> that could grow large over time, unless you specifically disabled trace files by setting <code>oracleXATrace="false"</code> in <code>config.xml</code>.</p> <p>The code was modified to set the default value of <code>oracleXATrace</code> to false if no value is specified.</p>
CR125705	<p>When using a JDBC MultiPool, WebLogic Server threw a resource exception to the client and failed to serve connections from a backup pool if the initial pool was fully reserved at the time of the connection attempt. The code was modified to throw a <code>ConnectDeadException</code> instead, which the MultiPool interprets as a reason to fail over to the next pool in its list.</p>
CR127891	<p>The format of the connection leak file was modified to make the information more readable.</p>

## jDriver

Change Request Number	Description
CR104968	<p>WebLogic jDriver for Oracle/XA did not properly handle the NLS_NUMERIC_CHARACTERS parameter set by an Oracle RDBMS instance. This caused a <code>java.sql.SQLException</code> when retrieving double or float values if a comma (',' ) was used as a decimal separator in the database:</p> <pre>java.sql.SQLException: java.lang.NumberFormatException - '1,2'     at weblogic.jdbc.oci.ResultSet.getFloat(ResultSet.java:312)     at weblogic.jdbc.jta.ResultSet.getFloat(ResultSet.java:190)     at weblogic.jdbc.rmi.internal.ResultSetImpl.getFloat(ResultSetImpl.java:224)     at weblogic.jdbc.rmi.internal.ResultSetStraightReader.getFloat(ResultSetStraightReader.java:67)     at weblogic.jdbc.rmi.SerialResultSet.getFloat(SerialResultSet.java:219)     at examples.servlets.DBAccess.service(DBAccess.java:54)     at [...]</pre> <p>The problem did not occur with the non-XA version of WebLogic jDriver for Oracle. This problem was solved with a code fix.</p>
CR107474	<p>In WebLogic Server 6.1 Service Pack 3, <code>OciObjectStream</code> did not nullify a buffer after a <code>close()</code> was performed. This problem was solved with a code fix.</p>
CR113462	<p>WebLogic Server Service Pack 5 threw a <code>NumberFormatException</code> when using <code>BigDecimal</code> types in an environment with Oracle's NLS_LANG setting. The problem did not occur when using Oracle's NLS_NUMERIC_CHARACTERS parameter to match American-style numeric character definitions. This problem was solved with a code fix.</p>

# JMS

Change Request Number	Description
CR103001	<p>WebLogic Server displayed an overly long exception if a server hosting a Messaging Bridge destination was unavailable. The code was fixed to display a shorter exception message.</p>
CR104538	<p>A JMS deadlock was fixed by changing the <code>BESession.java close(long lastSequenceNumber)</code> method. The deadlock could be viewed in the partial thread dump:</p> <pre data-bbox="327 578 1163 1446"> ExecuteThread: '9' for queue: 'queue_1'" daemon prio=5 tid=0x2757938 nid=0x5e waiting for monitor entry [0xce97f000..0xce97fc68] at weblogic.jms.backend.BETopic._addMessageToConsumers(BETopic.java:590) at weblogic.jms.backend.BETopic.addMessageToConsumers(BETopic.java:296) at weblogic.jms.backend.BEMessageWriteListener.storeIOComplete(BEMessageWriteListener.java:85) at weblogic.jms.store.StoreRequest.execute(StoreRequest.java:342) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)  "ExecuteThread: '14' for queue: 'queue_2'" daemon prio=5 tid=0x57b200 nid=0x54 waiting for monitor entry [0xd1b7f000..0xd1b7fc68] at weblogic.jms.backend.BESession.stop(BESession.java:386) at weblogic.jms.backend.BESession.close(BESession.java:434) at weblogic.jms.backend.BESession.close(BESession.java:1083) at weblogic.jms.backend.BESession.invoke(BESession.java:1024) at weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Request.java:585).... </pre>

<b>Change Request Number</b>	<b>Description</b>
CR112750	<p>When the server failed to send JMS messages, there was no error message on the client. The transaction manager had declared the messages unhealthy, and JMS was rolling back the messages when it failed to enlist resources.</p> <p>The failure to enlist the transaction is now communicated back to client, which will be able to report that the commit failed.</p>
CR120619	<p>When you removed all targets from a JMS Server having a JMS Template and then tried to re-target the JMS Server, WebLogic Server threw an exception:</p> <pre>&lt;Aug 13, 2003 4:50:58 PM PDT&gt; &lt;Error&gt; &lt;JMS&gt; &lt;Failed to deploy JMS Server "server_name" due to weblogic.jms.common.JMSEException: Error initializing JMSServer server_name. weblogic.jms.common.JMSEException: Error initializing JMSServer server_name     at weblogic.jms.backend.BackEnd.initialize(BackEnd.java:448)     at weblogic.jms.JMSService.createBackEnd(JMSService.java:906)     at weblogic.jms.JMSService.addJMSServer(JMSService.java:1273)     at weblogic.jms.JMSService.addDeployment(JMSService.java:1169)     at weblogic.management.mbeans.custom.DeploymentTarget.addDeploy ment(DeploymentTarget.java:364)     at weblogic.management.mbeans.custom.DeploymentTarget.addDeploy ment(DeploymentTarget.java:150)     at java.lang.reflect.Method.invoke(Native Method) [...]</pre> <p>This problem occurred because the JMS service exported a temporary destination factory to the RMI runtime, and the factory reference was not removed when the factory was unbound from JNDI. The code was fixed so that the reference to the temporary destination factory is now removed when the factory is unbound. Untargeting and retargeting a JMS Server having a JMS Template no longer causes an exception.</p>
CR122749	<p>Fixed a problem where calling EJBComponentRuntimeMBean.getEJBRuntimes() caused an ASSERTION FAILED error followed by an ArrayStoreException.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR123675	<p>A problem in the optimization code for non-durable messages sometimes caused a destination to be nullified. This would result in the following exception while paging out messages under heavy loads:</p> <pre>&lt;Sep 12, 2003 9:54:12 PM EDT&gt; &lt;Error&gt; &lt;Kernel&gt; &lt;BEA-000802&gt; &lt;ExecuteRequest failed java.lang.NullPointerException. java.lang.NullPointerException at weblogic.jms.common.MessageImpl.writeExternal(MessageImpl.java:1622) at weblogic.jms.common.TextMessageImpl.writeExternal(TextMessageImpl.java:92) at weblogic.jms.store.ObjectIOBypassImpl.writeObject(ObjectIOBypassImpl.java:155) at weblogic.jms.store.BufferDataOutputStream.writeObject(BufferDataOutputStream.java:175) at weblogic.jms.store.FileIOStream.write(FileIOStream.java:506) at weblogic.jms.store.StoreRequest.doTheIO(StoreRequest.java:282) at weblogic.jms.store.JMSStore.execute(JMSStore.java:493) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:197) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:170)</pre> <p>The problem was solved with a code fix.</p>
----------	---

---

## JNDI

Change Request Number	Description
CR105592	<p>In WebLogic Server 6.1 Service Pack 5 the server attempted to pop the environment from a ReadOnlyWrapper that did not push the environment onto a thread. This caused the following AssertionError and EmptyStackException:</p> <pre> java.util.EmptyStackException     at weblogic.utils.collections.Stack.pop(Stack.java:82)     at weblogic.kernel.ResettableThreadLocalStack.pop(ResettableThreadLocalStack.java:79)     at weblogic.jndi.internal.ThreadEnvironment.pop(ThreadEnvironment.java:18)     at weblogic.jndi.internal.WLContextImpl.close(WLContextImpl.java:72)     at weblogic.jndi.factories.java.ReadOnlyContextWrapper.close(ReadOnlyContextWrapper.java:30) [... ] ----- nested within: ----- weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[ attempt to pop from an empty stack ] - with nested exception: [java.util.EmptyStackException]     at weblogic.kernel.ResettableThreadLocalStack.pop(ResettableThreadLocalStack.java:81)     at weblogic.jndi.internal.ThreadEnvironment.pop(ThreadEnvironment.java:18)     at weblogic.jndi.internal.WLContextImpl.close(WLContextImpl.java:72) [... ] </pre> <p>The code was modified so that a pop is not attempted on a ReadOnlyWrapper.</p>
CR105761	<p>This Service Pack includes a new JNDI caching mechanism that significantly improves the performance of JNDI lookups in the server.</p>

---

Change Request Number	Description
-----------------------	-------------

---

CR100726	WebLogic Server now properly logs a <code>ClassNotFoundException</code> if you target a stateless session EJB to a single member of a WebLogic Server cluster and you fail to set <code>home-is-clusterable</code> and <code>stateless-bean-is-clusterable</code> to false in the deployment descriptor.
----------	--

---

CR110916	<p>WebLogic Server threw an <code>EmptyStackException</code> if a context was created in an <code>ejbCreate()</code> method and <code>Context.close()</code> was subsequently called in an <code>ejbRemove()</code> method. The partial exception text is:</p> <pre>java.util.EmptyStackException     at weblogic.utils.collections.Stack.pop(Stack.java:82)     at weblogic.kernel.ResettableThreadLocalStack.pop(ResettableThreadLocalStack.java:79)     at weblogic.jndi.internal.ThreadEnvironment.pop(ThreadEnvironment.java:18)     at weblogic.jndi.internal.WLContextImpl.close(WLContextImpl.java:72)     at javax.naming.InitialContext.close(InitialContext.java:476)     [...]</pre> <p>This problem was solved with a code fix.</p>
----------	--

---

# JSP

---

<b>Change Request Number</b>	<b>Description</b>
CR088525	<p>In previous WebLogic Server 6.1 Service Packs, JSP parameters with a value of "/////" were interpreted by <code>getParameter()</code> as having the value "/".</p> <p>The problem was resolved with a code change.</p>
CR092039	<p>WebLogic Server threw an <code>UnsupportedEncodingException</code> if you included extra quotation marks around the <code>charset</code> value of a JSP. For example, the following tag would yield the exception:</p> <pre>&lt;%@ page contentType="text/html; charset="\Shift_JIS\" " %&gt;</pre> <p>The problem was resolved with a code fix.</p>
CR093014	<p>In WebLogic Server 6.1 SP02, Java comments that start in one scriptlet and end in the next one, cause this exception:</p> <pre>&lt;22.07.2002 14:12:24 CEST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;[WebAppServletContext(7422744,DefaultWebApp,/DefaultWebApp) ] Servlet failed with Exception weblogic.servlet.jsp.JspException: (line 12): scriptlet close brace '}' unbalanced at line 12 which breaks scope '_base_service_scope_' at weblogic.servlet.jsp.ScriptletScopeLexer.mCLOSE_BRACE(ScriptletScopeLexer.java:527) at weblogic.servlet.jsp.ScriptletScopeLexer.mTOKEN(ScriptletScopeLexer.java:232) at weblogic.servlet.jsp.ScriptletScopeLexer.nextToken(ScriptletScopeLexer.java:159) at weblogic.servlet.jsp.ScriptletScopeLexer.parse(ScriptletScopeLexer.java:119) at weblogic.servlet.jsp.JspLexer.mSCRIPT(JspLexer.java:4367) at weblogic.servlet.jsp.JspLexer.mSTANDARD_THING(JspLexer.java:2173) at ....</pre> <p>The problem was resolved by a code fix to the make the <code>ScriptletScopeLexer</code> to skip an entire Java comment.</p>

---

---

Change Request Number	Description
CR093520	<p>When you precompiled JSPs on a machine in one time zone, and then deployed those same JSPs on a server in a different time zone, WebLogic Server sometimes re-compiled the JSPs. This occurred because WebLogic Server checked JSPs by comparing the local timestamp of the JSPs (as embedded by the JAR utility) against the timestamps in the generated class files.</p> <p>The problem was resolved by storing the time zone at compile time and using that time zone at deployment time to determine whether recompilation is necessary.</p>
CR100823	<p>If you refreshed a JSP with a copy, and the copy did not parse correctly, WebLogic Server entered a deadlock condition. The problem involved two separate deadlocks. The deadlocks were removed by throwing and evaluating an exception in the JSP stub level, and by removing unnecessary synchronization of threads in <code>getJarFiles()</code>.</p>
CR102628	<p>The following JSP code:</p> <pre data-bbox="327 703 1163 950">&lt;jsp:plugin type="applet" code="examples.applets.PhoneBook1.class" codebase="/bea_WebLogic Server_internal/classes/DefaultWebApp@DefaultWebApp/" height="800" width="500" type="applet" jreversion="1.3.1_06" nspluginurl="http://java.sun.com/products/plugin/1.1.3/plugi n-install.html"; iepluginurl="http://java.sun.com/products/plugin/1.1.3/ jinstall-113-win32.cab#Version=1,1,3,0" &gt;</pre> <p>generated this HTML:</p> <pre data-bbox="327 998 1163 1161">&lt;embed type="application/x-java-applet;version=1.3.1_06" pluginspage="http://java.sun.com/products/plugin/1.1.3/plugi n-install.html"; height="800" width="500" code="examples.applets.PhoneBook1.class" codebase="/bea_WebLogic Server_internal/classes/DefaultWebApp@DefaultWebApp/" &gt;</pre> <p>The generated HTML code failed on Netscape: Netscape attempted to download the plug-in from WebLogic Server instead from the SUN site.</p> <p>When the plugins page line was moved before the codebase line, the code worked correctly on Netscape.</p> <p>A code fix to order the applet attributes properly resolved the problem.</p>

---

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR103214	<p>When you set the <code>compilerclass</code> JSP parameter, WebLogic Server logged a <code>compilerclass=null</code> message in the startup log, as in:</p> <pre>JSPServlet with initArgs '[JspConfig: verbose=true,packagePrefix=jsp_servlet,-compiler= C:\61sp4\jdk131/bin/javac,compileFlags=,workingDir=C:\61sp4\ wlserver6.1\config\examples\applications\ examplesWebApp\WEB-INF\_tmp_war_examples_examplesWebApp,page CheckSeconds=1,superclass=weblogic.servlet.jsp. JspBase,keepgenerated=false,precompileContinue=false, compilerSupportsEncoding=true,encoding=null,defaultfilename= index.jsp,compilerclass= null,noTryBlocks=false]'</pre> <p>The problem occurred because the setting of <code>compilerclass</code> was not used during startup. The correct setting was used for compiling JSPs. The code was fixed to obtain the correct parameter value during server startup.</p>
----------	---

---

Change Request Number	Description
CR105229	<p>WebLogic Server could fail to evaluate empty BodyTags, causing the exception:</p> <pre> &lt;ExecuteThread: '14' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;WLS Kernel&gt;&gt; &lt;&gt; &lt;BEA-101017&gt; &lt;[ServletContext(id=12216222,name=xmlxTestApp,context-path=/xmlxTestApp)] Root cause of ServletException. javax.servlet.jsp.JspException: Could not find file: java.io.FileNotFoundException: Could not find appropriate xml file at weblogicx.xml.tags.XsltTag.doEndTag(XsltTag.java:280) at jsp_servlet.__xsltprocessor._jspService(__xsltprocessor.java :195) at weblogic.servlet.jsp.JspBase.service(JspBase.java:33) at weblogic.servlet.internal.ServletStubImpl\$ServletInvocationA ction.run(ServletStubImpl.java:1053) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:387) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv letStubImpl.java:305) at weblogic.servlet.internal.WebAppServletContext\$ServletInvoca tionAction.run(WebAppServletContext.java:6304) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(Aut henticatedSubject.java:317) at weblogic.security.service.SecurityManager.runAs(SecurityMana ger.java:118) at weblogic.servlet.internal.WebAppServletContext.invokeServlet (WebAppServletContext.java:361 [...] This problem was solved with a code fix.</pre>

<b>Change Request Number</b>	<b>Description</b>
CR105772	<p>WebLogic Server displayed a JSP parsing error if a JSP used a tag library that specified an empty body tag using an empty element tag, as in:</p> <pre>&lt;%@taglib uri="/problem-taglib" prefix="c" %&gt; &lt;html&gt; &lt;head&gt;&lt;title&gt;Problem scenario&lt;/title&gt;&lt;/head&gt; &lt;body&gt; &lt;table border="1"&gt;&lt;tr&gt;&lt;th&gt;Local file content&lt;/th&gt; &lt;th&gt;Included file content&lt;/th&gt;&lt;/tr&gt; &lt;tr&gt;   &lt;td&gt;Cell containing data from the original JSP&lt;/td&gt;   &lt;td&gt;&lt;c:import url="IncludedResource.jsp" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt;</pre> <p>The problem did not occur if a closing element tag was added to the empty body content, as in:</p> <pre>&lt;td&gt;Cell containing data from the original JSP&lt;/td&gt;   &lt;td&gt;&lt;c:import url="IncludedResource.jsp" &gt;&lt;/c:import&gt;&lt;/td&gt;</pre> <p>This problem was solved with a code fix.</p>
CR106072	<p>The <code>pageCheckSeconds</code> attribute, which sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling, did not work the first time a JSP was modified.</p> <p>This problem occurred because WebLogic Server did not set a <code>lastStaleCheck</code> time the first time a JSP was invoked.</p> <p>The code was modified so that if <code>lastStaleCheck</code> is not set yet, it is set to the current time. This prevents the JSP from being recompiled unnecessarily.</p>
CR107042	<p>If you specified <code>com.sun.tools.javac.Main</code> as the value of the JSP <code>compilerclass</code> parameter, WebLogic Server would suddenly exit when you accessed the JSP. This problem was solved with a code fix.</p>

<b>Change Request Number</b>	<b>Description</b>
CR111423	<p>When packaged in a WAR file, precompiled JSPs stored in a sub-context of the application were always recompiled upon deployment to WebLogic Server. This problem did not occur when JSPs were deployed from an exploded archive directory, or for JSPs in the root-context of a WAR file.</p> <p>The problem was caused because of a difference in the rounding behavior of timestamps used in the <code>jar</code> and <code>zip</code> formats. The discrepancy in rounding could cause an older timestamp (by one second) to be recorded in class files inside the WAR file, triggering the server to recompile the classes.</p> <p>The code was modified to advance the timestamps in compiled JSP classes by one second, thereby preventing JSPs from being recompiled.</p>
CR120914	<p>When you used the WebLogic Server form validation tag library, request parameters were not available to subsequent JSPs. This problem was solved with a code fix.</p>
CR127515	<p>In WebLogic Server 6.1 SP02 a <code>javac</code> error message through <code>weblogic.jspc</code> was corrupt in Japanese installation. The problem was corrected by a code change to use <code>InputStreamReader</code> instead of <code>DataInputStream</code>.</p>

# JTA

---

<b>Change Request Number</b>	<b>Description</b>
CR082504	<p>When using the Sybase jConnect/XA Driver, connections were not reusable after being returned to the connection pool. This would cause the following exception, after all available connections in the pool were used:</p> <pre>XA error: XAER_RMERR : A resource manager error has occurred in the transaction branch start() failed on resource 'ZeusPool': XAER_RMERR : A resource manager error has occurred in the transaction branch javax.transaction.xa.XAException</pre> <p>The problem occurred because Sybase initiates a local transaction for DDL calls, and the transaction was not cleaned up when the connection was returned to the pool.</p> <p>The connection cleanup code was fixed to end the local Sybase transaction generated by DDL calls.</p>
CR091192	<p>This Service Pack includes an enhancement to recover transactional resource managers immediately, rather than wait five minutes before recovery.</p>

---

<b>Change Request Number</b>	<b>Description</b>
CR091974	<p>If XA recovery did not return an XID, WebLogic Server 6.1 Service Pack 3 could throw a <code>NullPointerException</code>:</p> <pre>java.lang.NullPointerException     at weblogic.transaction.internal.ServerResourceInfo.rollback(ServerResourceInfo.java:721)     at weblogic.transaction.internal.ServerSCInfo.rollback(ServerSCInfo.java:318)     at weblogic.transaction.internal.ResourceDescriptor.rollbackXids(ResourceDescriptor.java:1160)     at weblogic.transaction.internal.ResourceDescriptor.recover(ResourceDescriptor.java:1060)     at weblogic.transaction.internal.ResourceDescriptor.access\$9(ResourceDescriptor.java:1029)     at weblogic.transaction.internal.ResourceDescriptor\$1.execute(ResourceDescriptor.java:770)     at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)     at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)</pre> <p>The code was modified to check for the presence of an XID and prevent the exception from occurring.</p>
CR098273	<p>Stale entries in the transaction log sometimes caused unnecessary recovery overhead when a server is restarted. This problem was solved with a code fix.</p>

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR103327	<p>After a transaction has timed out, two threads could try to roll back the same transaction ID, resulting in an error similar to:</p> <pre>XA.end FAILED (rm=pool_name, xar=pool_name, error code: XAER_RMERR : A resource manager error has occurred in the transaction branch, message: null&gt; oracle.jdbc.xa.OracleXAException   at oracle.jdbc.xa.OracleXAResource.checkError(OracleXAResource. java:659)   at oracle.jdbc.xa.client.OracleXAResource.end(OracleXAResource. java:305)   at weblogic.jdbc.jta.VendorXAResource.end(VendorXAResource.java :47)</pre> <p>This problem was solved with a code fix.</p>
----------	--

---

CR103601	<p>In WebLogic Server 6.1 SP03 and SP04, if an object passed in the “equals” method of <code>weblogic.transaction.internal.XidImpl</code> is not an instance of <code>XidImpl</code> (for instance, it is of type <code>String</code>), then this exception is thrown:</p> <pre>java.lang.ClassCastException: java.lang.String at weblogic.transaction.internal.XidImpl.equals(XidImpl.java:11 4) at test.main(test.java:9)</pre> <p>The problem was solved with a code fix to check and report type mismatches.</p>
----------	--

---

Change Request Number	Description
CR113226	<p>When a resource name contained more than 64 characters, WebLogic Server 6.1 Service Pack 4 could throw the following exception when testing the connection pool:</p> <pre>java.sql.SQLException: XA error: XAER_RMERR : A resource manager error has occurred in the transaction branch start() failed on resource 'weblogic.jdbc.jta.DataSource' null at weblogic.jdbc.jta.DataSource.enlist(DataSource.java:1167) at weblogic.jdbc.jta.DataSource.refreshXAConnAndEnlist(DataSource.java:1133) at weblogic.jdbc.jta.Connection.getXAConn(Connection.java:153) at weblogic.jdbc.jta.Connection.prepareStatement(Connection.java:241) [...]</pre> <p>The problem occurred because only the first 64 characters were tested for uniqueness. The code was modified to properly handle resource names longer than 64 characters.</p>
CR126201	<p>In a multiple-server domain, if a Managed Server was rebooted to use a different address or port number, the JTA subsystem failed to update the address information. This would cause the following exception when the changed server was rebooted:</p> <pre>javax.naming.CommunicationException. Root exception is java.net.ConnectException: t3://ip_address:port_number: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused; No available router to destination [...]</pre> <p>The code was fixed to obtain new address information from the Administration Server in response to an address or port change.</p>

## Node Manager

---

<b>Change Request Number</b>	<b>Description</b>
CR104285	Node Manager's shared object code could cause a segment violation if certain code paths were taken while starting a server instance. The same code paths also failed when using IBM's zLinux JDK. These problems were solved with a code fix to Node Manager.
CR125829	Please review the security advisory information at <a href="http://dev2dev.bea.com/resource/library/advisories/notifications/BEA03_42.00.jsp">http://dev2dev.bea.com/resource/library/advisories/notifications/BEA03_42.00.jsp</a> .

---

## OA&M (Operations, Administration, and Management)

---

<b>Change Request Number</b>	<b>Description</b>
CR090118	<p>In WebLogic Server 6.1 SP03 and later, temporary directories created by the Administration Console deployment descriptor editor in under the <code>bea\wlserver6.1</code> directory with name <code>jarxxxx</code> (for example, <code>bea\wlserver6.1\jar7018</code>) were not deleted.</p> <p>Analysis revealed that the temporary directories were not deleted because of a open file stream on an inner file.</p> <p>The problem was resolved with a code change.</p>

---

<b>Change Request Number</b>	<b>Description</b>
CR093653	<p>In WebLogic Server 6.1 SP04, updates to an exploded web application that was deployed to a Managed Server were reflected in the exploded application staged on the Managed Server, but not in the .war file in the .wlnotdelete directory. After server restart the old version of the application was deployed.</p> <p>Refreshing the application using <code>weblogic.refresh</code> caused the updated .jsp to be copied into the exploded application format staged on the Managed Server, but not the original version of the .war.</p> <p>The problem was resolved by a code change to remove the application entry from the local deployment file.</p> <p>As a result of this change, the latest version of an application is copied from its source to a Managed Server's staging area when the Managed Server is restarted after a refresh.</p>
CR093687	<p>In WebLogic Server SP03, a startup class that created a dynamic connection pool and targets it to a server instance, caused the server to hang, when <code>LoadBeforeAppDeployments=true</code> for startup class. No exception was reported. Analysis revealed a synchronization problem, which was corrected with a code fix.</p>
CR095967	<p>In WebLogic Server SP03 and SP04, when a Managed Server and Administration Server were shutdown at the same time, the Managed Server tried to reconnect to the Administration Server.</p> <p>The problem was solved with a code change to ensure that Managed Servers do not attempt to reconnect to the Administration Server while in shutdown or suspend modes.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR097152	<p>In WebLogic Server SP03 and SP04, restarting the Administration Server resulted in the error:</p> <pre>Admin restart causes java.rmi.ConnectException: This RJVM has already been shutdown.</pre> <p>The issue was consistently reproduced using these steps:</p> <ol style="list-style-type: none"><li>1. Configure a cluster such that the cluster instances on at least two machines.</li><li>2. Deploy a sample EJB and target to cluster.</li><li>3. Now restart the Admin server.</li><li>4. Now go to console and untarget the EJB from cluster, apply and then retarget the EJB to cluster and Apply the changes.</li></ol> <p>The problem occurred because <code>weblogic.management.internal.MBeanProxy</code> was using a stale stub of <code>AdminMBeanHome</code>, because the Administration Server was restarted and the <code>AdminMBeanHome</code> object instance (Dynamic Proxy Stub) was no longer valid.</p> <p>The problem was solved with a code change.</p>
----------	--

---

CR099973	<p>In WebLogic Server 6.1 Service Pack 3, using the <code>UpdateLicense</code> utility with the <code>license_update_file</code> failed to properly update the license file. This lead to the following exception when booting the server:</p> <pre>\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ License Exception \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$</pre> <pre>Unable to start WebLogic Server !!</pre> <pre>WebLogic: license signature validation error!</pre> <p>The problem was solved with a code fix.</p>
----------	---

---

Change Request Number	Description
CR102593	<p>In WebLogic Server 6.1 SP04, a <code>ClassCastException</code> occurred when retrieving <code>ServletSessionRuntimeMBean</code>.</p> <p>The problem was exhibited in a two-node cluster, on which an application retrieves the array of <code>ServletSessionRuntimeMBeans</code> from the <code>WebAppComponentRuntime</code> of each cluster node, for use in determining whether a particular user has already logged on to one of the two nodes.</p> <p>The call sequence is as follows:</p> <ol style="list-style-type: none"><li>1. call:node1 ---&gt; test.jsp (HTTP Session is created)</li><li>2. call:node2 ---&gt; submit data from test.jsp (via GET) to /ctrl/* Servlet</li></ol> <p>Repeating this call sequence several times in the same HTTP session leads to a <code>ClassCastException</code> in the <code>WebAppComponentRuntime.getServletSessions()</code> method:</p> <pre>ClassCastException: java.lang.ClassCastException: \$Proxy79 at weblogic.servlet.internal.session.SessionData.getServletSessionRuntimeMBean(SessionData.java:271) at weblogic.servlet.internal.session.SessionContext.getServletSessionRuntimeMBean(SessionContext.java:199) at weblogic.servlet.internal.session.SessionContext.getServletSessionRuntimeMBeans(SessionContext.java:216) at weblogic.servlet.internal.WebAppServletContext.getServletSessionRuntimeMBeans(WebAppServletContext.java:2442) at weblogic.servlet.internal.WebAppRuntimeMBeanImpl.getServletSessions(WebAppRuntimeMBeanImpl.java:126) at java.lang.reflect.Method.invoke(Native Method) at...</pre> <p>The problem occurred because <code>ServletSessionRuntimeMBean</code> was cast incorrectly to the implementation—<code>ServletSessionRuntimeMBeanImpl</code> in <code>SessionData.java</code>.</p> <p>The problem was solved with a code change.</p>
CR102860	<p>This release solves a JMX memory leak. The leak was detected running load tests on WebLogic Server 6.1 SP02. When running JMX for several days, the heap consumption after garbage collection grew from 17 to 40 MB.</p>

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR103735	<p>In WebLogic Server SP03, if the domain and a cluster in the domain have the same name, duplicate entries appear for the Targets attribute in the domain's <code>config.xml</code> file. The following was observed:</p> <p>When an application was first targeted to the cluster through the console, there were no problems, and the target was correctly reflected in <code>config.xml</code>.</p> <p>After restarting the Administration Server, the Administration Console did not show that the application was targeted to the cluster, and a Managed Server in the cluster did not show the application as bound in its JNDI tree. The target was still correctly reflected in <code>config.xml</code>.</p> <p>After the user re-targeted the application to the cluster using the Administration Console, <code>config.xml</code> showed the cluster name twice in its Targets attribute.</p> <pre>&lt;Application Deployed="true" Name="seb" Path=". \config\rom\applications"&gt; &lt;WebAppComponent Name="seb" Targets="rom,rom" URI="seb.war"/&gt; &lt;/Application&gt;</pre> <p>The problem occurred because, in any version of WebLogic Server, all targets domains, clusters, servers, virtual hosts) must have unique names.</p> <p>The problem was solved with a code change to modify <code>tweblogic.managment.internal.UnresolvedMBean.java</code> to ignore the <code>DomainMBean</code> when resolving in the case where the attribute is a target.</p>
----------	---

---

Change Request Number	Description
CR105338	<p>In WebLogic Server SP03, WebLogic Server stops logging, on both Administration Server and Managed Servers, when the maximum number of files is reached.</p> <p>After the following error messages are printed out to the domain log file, the server logging service is shutdown.</p> <pre>####&lt;Apr 21, 2003 3:17:39 PM GMT+09:00&gt; &lt;Alert&gt; &lt;Log Management&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;BEA-170017&gt; &lt;The log file .\myserver\myserver.log will be rotated. Reopen the log file if tailing has stopped. This can happen on some platforms like Windows.&gt; ####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Critical&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised several exceptions. Shutting it down&gt; ####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Error&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised exception when opening. Exception weblogic.logging.LogRotationException null&gt; ####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Error&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised exception when opening. Exception weblogic.logging.LogRotationException null&gt;</pre> <p>If the problem occurs, all log messages are not printed out to the server log file. This problem occurs on the admin server and the managed server</p> <p>Analysis revealed that WebLogic Server closed log files during rotation and, if log rotation failed, the log files would remain closed.</p> <p>Log Rotation failed when WebLogic Server generated the wrong index for the new log file. Since it assumed that the file that had the latest time stamp is the latest index. It never checked to see if a file with the index already existed. WebLogic Server did not attempt to delete the log file if the log file already existed and did not check to see if the rename( ) was successful.</p> <p>In addition, for time based log rotation, on multi-CPU machines, multiple rotations happened within the same millisecond. The problems were changed with code fixes.</p>

---

**Change Request Number**    **Description**

---

CR108448            In WebLogic Server 6.1 SP05 restart of a Managed Server resulted in a warning. The Administration Server and Managed Server were running, and the Managed Server was shutdown. The Administration Server showed that the Managed Server was not running. Upon restarting the Managed Server, this warning was issued:

```
<Warning> <Management> <c4wlg001> <node001> <ExecuteThread:
'1' for queue: '__weblogic_admin_rmi_queue'> <system> <>
<141029> <Unable to reconnect to Admin Server running at
http://c4wlg001:7001 from Managed Server - node001.>
weblogic.management.configuration.ConfigurationException:
Server: node001 is already running. at
weblogic.management.Admin.registerManagedHome2AdminHome(Admin.java:1679) at
weblogic.management.Admin.reconnectToAdminServer(Admin.java:1637) at
weblogic.t3.srvr.ServerRuntime.reconnectToAdminServer(ServerRuntime.java:413) at
java.lang.reflect.Method.invoke(Native Method) at
weblogic.management.internal.DynamicMBeanImpl.invokeLocally(DynamicMBeanImpl.java:636) at
weblogic.management.internal.DynamicMBeanImpl.invoke(DynamicMBeanImpl.java:621) at...
```

The Managed Server starts successfully.

Analysis revealed that an error check was performed at an inappropriate point, preventing the JNDI tree to be appropriately updated. The problem was corrected by a code fix.

---

CR108727            In WebLogic Server SP04, running with CR071109\_610sp2.jar, an Administration Server started with the command-line options:

```
weblogic.management.discover.interval = 60
weblogic.management.discover.retries = 6
weblogic.management.internal.debug = true
```

after a failure, (java.rmi.ConnectException is thrown) managed servers were discovered and re-connected successfully, but an `OutOfMemoryError` occurred.

Subsequently, MBean invocations on managed servers failed with this exception:

```
java.rmi.NoSuchObjectException: RemoteInvokable - id: '267'
```

Analysis revealed that `ManagedServerRediscoveryChecker` thread had stopped running, causing Managed Servers not to be discovered. The problem was solved with a change to retry logic to ensure that the discovery thread always runs when needed.

---

---

Change Request Number	Description
-----------------------	-------------

---

CR127860	<p>In WebLogic Server 6.1 SP05, shutdown classes were not called on Managed Servers when the Administration Server was not running. The following exception was thrown:</p> <pre> &lt;29.10.2003 10:48:16 CET&gt; &lt;Notice&gt; &lt;WebLogicServer&gt; &lt;Started WebLogic Managed Server "managed1" for domain "mydomain" running in Production Mode&gt; &lt;29.10.2003 10:49:54 CET&gt; &lt;Alert&gt; &lt;WebLogicServer&gt; &lt;The disabling of server logins has been requested by system&gt; &lt;29.10.2003 10:49:54 CET&gt; &lt;Alert&gt; &lt;WebLogicServer&gt; &lt;Server logins have been disabled.&gt; &lt;29.10.2003 10:49:54 CET&gt; &lt;Alert&gt; &lt;WebLogicServer&gt; &lt;Server shutdown has been requested by system&gt; &lt;29.10.2003 10:49:54 CET&gt; &lt;Alert&gt; &lt;WebLogicServer&gt; &lt;The shutdown sequence has been initiated.&gt; &lt;29.10.2003 10:49:56 CET&gt; &lt;Emergency&gt; &lt;Server&gt; &lt;Unable to initialize the server: 'Fatal initialization exception Throwable: weblogic.rmi.extensions.RemoteRuntimeException - with nested exception: [java.rmi.ConnectException: Could not establish a connection with 4622510611903127260S:172.23.64.209:[7001,7001,7002,7002,7001 ,7002,-1]:mydomain:myserver, java.rmi.ConnectException: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused: connect; No available router to destination] java.rmi.ConnectException: Could not establish a connection with 4622510611903127260S:172.23.64.209:[7001,7001,7002,7002,7001 ,7002,-1]:mydomain:myserver, java.rmi.ConnectException: Destination unreachable; nested exception is:     java.net.ConnectException: Connection refused: connect; No available router to destination     at weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java:276)     at weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java:428).. The problem was corrected with a code change to run shutdown classes before shutting down the management layer.</pre>
----------	--

---

---

**Change Request Number**    **Description**

---

CR122538            If a JMX client application used `getAllMBeans()` on a WebLogic Server, exceptions similar to the following were thrown:

```
Exception in thread "main" java.lang.ClassNotFoundException:
weblogic.management.descriptors.ejb11.MailServiceMBean
    at
    java.net.URLClassLoader$1.run(URLClassLoader.java:195)
    at java.security.AccessController.doPrivileged(Native
Method)
    at
    java.net.URLClassLoader.findClass(URLClassLoader.java:183)
    at
    java.lang.ClassLoader.loadClass(ClassLoader.java:294)
    at
    sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:281
)
    at
    java.lang.ClassLoader.loadClass(ClassLoader.java:250)
    at
    java.lang.ClassLoader.loadClassInternal(ClassLoader.java:310
)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:190)
    at
    weblogic.management.internal.Helper.findClass(Helper.java:83
1)
[...]
```

----- nested within: -----  
weblogic.management.configuration.ConfigurationError - with  
nested exception:  
[java.lang.ClassNotFoundException:  
weblogic.management.descriptors.ejb11.MailServiceMBean]  
 at  
weblogic.management.internal.Helper.getAdminOrConfigMBeanInf  
o(Helper.java:118)  
[...]

This problem was solved with a code fix.

---

## Plug-ins

Change Request Number	Description
CR087204	[Apache] In WebLogic Server 6.1 SP03, <code>PathTrim</code> and <code>PathPrepend</code> added unexpected "/" at the end of URL. After <code>PathTrim</code> was applied to the original URL and it results to '/', an extra '/' was appended to the <code>PathPrepend</code> variable. This problem was solved with a code fix.
CR091910	[Apache] In WebLogic Server 6.1 SP03, the Apache plugin did not read <code>PathPrepend</code> when using <code>&lt;IfModule mod_weblogic.c&gt;</code> . The problem occurred with plugins for Apache 1.3.x and Apache 2.0.43. The problem was solved with a code fix.
CR092970	[Apache] In WebLogic Server 6.1 SP04, the Apache plugin caused sockets to remain in the <code>CLOSE_WAIT</code> state. Analysis revealed that WebLogic Server did not close the socket when returning a <code>sendRedirect</code> . The problem was solved with a code fix.
CR092327	[Apache] In WebLogic Server 6.1 SP04, setting the <code>WLogFile</code> did not work when <code>VirtualHosts</code> were used. The log file defaulted to <code>/tmp/wlproxy.log</code>
CR100070, CR107200	[Apache] When two virtual hosts are used in Apache server, each virtual host has defined its own <code>WLogFile</code> . The value of this parameter keeps changing among two <code>WLogFile</code> values defined for two virtual hosts. There were two problems: one, WebLogic Server was only setting <code>WLogFile</code> once upon server startup; two, WebLogic Server set it before getting it from <code>VirtualHost</code> . Setting <code>WLogFile</code> for each virtual host in the Apache plugin fixed the problems.
CR100601, CR105658, CR108747	(NSAPI) The plug-in that shipped with WebLogic Server 6.1 Service Pack 3 used the <code>WL-Proxy-Client-IP</code> header to send client IP addresses. This caused problems in heterogeneous environments, because earlier servers expected the plug-in to use the <code>X-Forwarded-For</code> and <code>Proxy-Client-IP</code> headers to transmit client IP addresses. The code was fixed to include both the <code>X-Forwarded-For</code> and <code>Proxy-Client-IP</code> headers, as well as <code>WL-Proxy-Client-IP</code> , so that heterogeneous environments can retrieve client addresses without modification.
CR101937	(NSAPI) If an HTTP request contained an <code>expect-continue</code> header, the plug-in would send a 100 (Continue) response even if the plug-in's client used HTTP/1.0. The code was fixed to comply with the HTTP/1.1 specification—the plug-in does not respond with a 100 (Continue) status if the client uses HTTP/1.0 or earlier (or if there was no <code>expect-continue</code> header in the request).

<b>Change Request Number</b>	<b>Description</b>
CR102616	(NSAPI, Apache) The plug-in now supports dynamic DNS lookups on a DNS name when the name returns a list of IP addresses from the DNS server and the plug-in is configured with <code>WebLogicHost = 'DNS name'</code> .
CR105123	<p>(ISAPI) In all versions of WebLogic Server, <code>DefaultFileName</code> does not work if <code>iisforward.dll</code> is not used. The problem is exhibited when:</p> <ul style="list-style-type: none"><li>■ Virtual Directory is configured.</li><li>■ Mime type is configured to * (proxy everything).</li><li>■ <code>DefaultFileName</code> is added to <code>iisproxy.ini</code></li></ul> <p>On a request for a directory that has no filename, the <code>DefaultFileName</code> is not used. The problem was corrected with a code fix.</p>
CR105173	<p>(WLS-NSAPI) In WebLogic Server 6.1 SP04, when a client stopped a response from being sent to it (for example, closing the browser before the response is completely received), a 500 [WRITE TOCLIENT ERROR] is logged in the webserver logs.</p> <p>This is unacceptable to our client who uses a webserver health monitoring tool to determine his server's health and this issue typically comes up when a request - &gt; response takes a fair amount of time.</p> <p>The webserver health monitoring tools use a 500 error to indicate that something is wrong with the server's health and that since this is not a server health issue but the client terminating the response, the error if any should not be 500.</p> <p>Request response path is as follows: client -&gt; proxyWebserver-&gt;plugin-&gt;wls and the expected response path is client &lt;- proxyWebserver&lt;-plugin&lt;-wls</p> <p>However, after the Weblogic server has successfully sent the response, but the webserver has not completely sent it to the client, the client aborts the communication and a 500 error is logged in the webserver's access.log which normally indicates that something is wrong with the server.</p> <p>The Customer is of the opinion that a different error or none at all should be logged for such a situation</p> <p>A code change was implemented so that 500 errors are not generated if the client breaks the connection.</p>

---

<b>Change Request Number</b>	<b>Description</b>
CR106764	(NSAPI) A thread in the plug-in could obtain a critical lock for a long duration (5 minutes by default, or configured using <code>WLIOTimeoutSecs</code> ), blocking all other threads and making WebLogic Server appear to hang. This problem was fixed with a code change to the plug-in.
CR107254	(Apache) Because Hewlett-Packard ceased to support the HP Apache-based Web Server version 1.3.x, WebLogic Server removed the Apache 1.3 plug-in for HP.
CR108092	(ISAPI) In WebLogic Server 6.1 Service Pack 4, the ISAPI plugin logged an unhandled exception error in the Windows event log when it encountered a modified cookie. The event text began with the line:  <code>The HTTP Server encountered an unhandled exception while processing the ISAPI Application.</code>  This problem was solved with a code fix to the ISAPI plugin.
CR109755	[Apache] The plug-in ignored configuration parameters that contained regular expressions other than wildcard characters (*/?). This could cause 404 errors to occur when using parameters such as:  <code>LocationMatch "/weblogic/(abc def)/ghi"</code>  This problem was solved with a code fix.
CR110664	(NSAPI) The plug-in code failed to catch an exception, which in turn caused iPlanet server to crash during the <code>sendResponse</code> phase. The plug-in code was changed to catch the exception.
CR111167	(ISAPI) In WebLogic Server 6.1 Service Pack 2, using the ISAPI plugin resulted in HTTP responses having two Date headers: one inserted by WebLogic Server and one inserted by IIS. This duplication of Date headers caused problems with certain caching services that expected a single Date header.  The problem was solved by updating the ISAPI plugin to filter out the Date header inserted by WebLogic Server.
CR113033	(ISAPI) In WebLogic Server 6.1 Service Pack 4, the plug-in did not recognize the <code>WLTempDir</code> flag for the <code>_wl_proxy</code> folder. The code was fixed to use the flag.

Change Request Number	Description
CR113093	<p>[Apache] When using multiple <code>MatchExpression</code> parameters in <code>httpd.conf</code> to route requests to different locations, as in:</p> <pre>MatchExpression *.jsp WebLogicHost=localhost WebLogicPort=8001 MatchExpression *.html WebLogicCluster=localhost:8001,localhost:8003</pre> <p>Each request overwrote the same global parameter info, which caused requests to go to the wrong location. In the above example, this problem resulted in <code>*.jsp</code> requests going to the server at port 8003.</p> <p>The code was fixed to ensure that each request uses its own copy of the parameter information.</p>
CR121341	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_39.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_39.00.jsp</a>.</p>
CR121688	<p>(Apache) The plug-in failed to parse a cookie if the exclamation character, “!”, was replaced by “%21” in a URL. This replacement is commonly done by WAP gateways when using URL rewriting. The code was fixed to correctly parse the characters in the URL.</p>
CR121943	<p>[Apache] The plug-in did not parse cookies when “%1” is replaced with “%21” in the URL. This replacement occurred when using a WAP gateway and URL rewriting. The problem was solved with a code fix.</p>
CR122207	<p>(NSAPI) If <code>KeepAliveEnabled</code> and <code>DynamicServerList</code> were both enabled, the plug-in could leave sockets in a <code>CLOSE_WAIT</code> state. This problem was solved with a code fix.</p>
CR122754	<p>(ISAPI) In WebLogic Server Service Pack 4, the plug-in parameter <code>WLExcludeByPathOrMimeType</code> did not work when forwarding by mime type. This problem was solved with a code fix.</p>
CR122755	<p>(ISAPI) In WebLogic Server Service Pack 4, the plug-in filter was bypassed if “.wforward” was manually appended to a URL. The code was modified to throw a 404 error if the initial request has a mime type of <code>.wforward</code>.</p>
CR123120, CR123775	<p>(Apache, NSAPI) If the POST method was used through the plug-in and the <code>Content-Length</code> was not defined, the proxy log file would contain message such as: POST and PUT requests *must* contain a <code>Content-Length</code></p> <p>The code was modified to set a content length of zero (0) if <code>Content-Length</code> is undefined.</p>

Change Request Number	Description
CR123925	<p>(ISAPI) The plug-in would sometimes respond to the browser with a 500 error message. This problem had three additional symptoms:</p> <ol style="list-style-type: none"> <li data-bbox="327 370 841 393">1. The IIS access log would show the message: <pre data-bbox="366 415 1126 464">Out-of-process+ISAPI+extension+request+failed. 500 1726 99122 2003 84078</pre> </li> <li data-bbox="327 490 926 513">2. The Windows Event Log would record Event ID 37: <pre data-bbox="366 539 1155 906">Event Type: Warning Event Source: W3SVC Event Category: None Event ID: 37 Date: 8/26/2003 Time: 6:45:03 PM User: N/A Computer: name Description: Out of process application '/LM/W3SVC/2/Root/caf' terminated unexpectedly. For additional information specific to this message please visit the Microsoft Online Support site located at: http://www.microsoft.com/contentredirect.asp.</pre> </li> <li data-bbox="327 932 723 954">3. The <code>wlproxy.log</code> entry showed: <pre data-bbox="366 980 1096 1107">Fri Nov 21 19:06:31 2003 Write to the browser failed: calling URL::close at line 1270 of .\iisproxy.cpp Fri Nov 21 19:06:31 2003 *****Exception type [WRITE_ERROR_TO_CLIENT] raised at line 1271 of .\iisproxy.cpp</pre> </li> </ol> <p>This problem was solved with a code fix.</p>
CR124433	<p>(ISAPI) If IIS was configured with <code>WlForwardPath=/</code>, the plug-in would try forward requests even if the server was down. The error page was never served to clients. The plug-in was modified to properly exclude paths in this situation.</p>
CR124464	<p>(NSAPI) A memory leak was detected that could cause the plug-in to crash. The problem occurred because the plug-in accessed an exception object after the object had been deleted. The code was fixed to retrieve the exception code from the exception object and then delete the object, which prevents the memory leak from occurring.</p>

Change Request Number	Description
CR125545	<p>(NSAPI) When a client stopped a response from being sent to it (for example, by closing the browser before the response had completed), the plug-in wrote a 500 [WRITE TO CLIENT ERROR] to the Web server log file. This could cause problems with health monitoring tools that interpret the 500 error as a problem in the Web server.</p> <p>This problem was solved with a code fix.</p>
CR125690	<p>(ISAPI) In a configuration that included nine IIS servers and nine clustered WebLogic Server instances, IIS crashed every a few hours, writing an Event 37 to the event log. The wlproxy log contained this message:</p> <pre>Thu Oct 09 13:01:46 2003 *****Exception type [WRITE_ERROR_TO_CLIENT] raised at line 1269 of .\iisproxy.cpp</pre> <p>Diagnosis revealed that the <code>Reader::fill()</code> method was not allocating enough memory while growing the initial buffer. 4 bytes to mark the end of buffer were getting lost and which resulted in the core dump. The problem was solved with a code fix.</p>
CR126103	<p>(NSAPI) During load testing, when NSAPI running on HP11.00 proxying to a 6 node cluster on 2 Solaris boxes (3 WebLogic Server instances on each), memory consumption steadily increased, and after approximately 50 minutes, the <code>ns-httpd</code> process crashed.</p> <p>The same load test did not crash on HP11.00 or Solaris.</p> <p>Analysis revealed that codes in <code>proxy.cpp</code> used <code>strdup()</code>, a native system call. <code>strdup()</code> allocates system memory to the program's heap space. WebLogic Server uses Iplanet's <code>FREE</code> macro to free previous allotted space when it is no longer needed. Because <code>FREE</code> does not free the allocated space by <code>strdup()</code> call, the memory leak occurred.</p> <p>The problem was solved by replacing all native <code>strdup()</code> system calls in <code>proxy.cpp</code> with Iplanet's <code>STRDUP</code> macro so the <code>FREE</code> macro knows what to free.</p>
CR126568	<p>(NSAPI) Plugin does not handle %0A in the post request gracefully</p> <p>A POST request %0A at the end sent to WebLogic Server through the NSAPI plug-in was not handled gracefully. The request added extraneous data into the body stream, and headers appeared at the end of the body. Requests sent directly to WebLogic Server were processed correctly it works fine. Customer needs a plugin which can handle %0A at the end gracefully.</p> <p>The problem was corrected by code change to the plug-in to detect and handle HTTP/0.9 responses correctly.</p>

---

Change Request Number	Description
-----------------------	-------------

---

CR126982	<p>(NSAPI) When WLEXcludePathOrMimeType was set, the file types were cut in the request to WebLogic Server, but iplanet failed to serve those files instead.</p> <p>For example, this request for a .jsp that contained a .jpg was made:</p> <pre>&lt;Object name="test5" ppath="*/weblogic/*"&gt; Service fn="wl_proxy" WebLogicHost="lorna" WebLogicPort="7001" PathTrim="/weblogic" Debug="ALL" DebugConfigInfo="ON" WLEXcludePathOrMimeType="*.jpg" &lt;/Object&gt;</pre> <p>The request for the .jsp was proxied to WebLogic Server, and the .jsp was displayed without the .jpg. Iplanet failed to server the jpg. The iplanet access log contained this message:</p> <pre>10.40.4.117 - - [28/Oct/2003:11:45:34 -0500] "GET /weblogic/images/logo_tm_onwt.jpg HTTP/1.1" 500 305 I get the following in wlproxy.log: Tue Oct 28 11:45:35 2003 ===== new request ===== Tue Oct 28 11:45:35 2003 INFO: SSL is not configured Tue Oct 28 11:45:35 2003 URI=[/hello.jsp] Tue Oct 28 11:45:35 2003 attempt #0 out of a max of 5 Tue Oct 28 11:45:35 2003 general list: trying connect to '10.40.4.117'/7001/7001 at line 1224 for '/hello.jsp' Tue Oct 28 11:45:35 2003 INFO: New NON-SSL URL Tue Oct 28 11:45:35 2003 Going to check the general server list Tue Oct 28 11:45:35 2003 WLS info : 10.40.4.117:7001 recycled? 0 Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept]=[*//*] Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept-language]=[en-us] Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept-encoding]=[gzip, deflate] Tue Oct 28 11:45:35 2003 Hdrs from Client:[user-agent]=[Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)]</pre> <p>WLEXcludePathOrMimeType should cause WebLogic Server to not service the request, and to pass control to the web server, allowing it to continue processing the request.</p> <p>The problem was solved with a code change.</p>
----------	--

---

# RMI

---

Change Request Number	Description
CR106281	<p>In WebLogic Server 6.1 SP04, replicated session beans under heavy load resulted in increasing heap usage and <code>OutOfMemoryErrors</code>.</p> <p>Analysis revealed that <code>examples.ejb20.basic.statefulSession.TraderBean_5ysgq2_EOImpl</code> objects were not being garbage-collected even when garbage collection was forced. For clustered stateful session beans, strong references were maintained to the <code>EOImpl</code> object primary in the <code>weblogic.rmi.cluster.PrimarySecondaryRemoteObject</code>. As a <code>remove</code> is never called on the bean, the reference to the <code>EOImpl</code> was never removed from the <code>eoMap</code>.</p> <p>A code fix was implemented to unexport the EO when the passivated bean has been deleted, after <code>session-timeout-seconds</code>.</p>

---

## RMI/IIOP

Change Request Number	Description
CR112991	<p>IIOP calls from a WebLogic Server 6.1 SP04 instance to an EJB running on a WebLogic Server 7.0 SP02 instance resulted in an exception when the IIOP connection idle timed out. The exception occurred on the WebLogic Server 7.0 side, when it tried to send back the response—the connection was closed. The exception is not sent to the 6.1 server, and the client thread hangs.</p> <p>The WebLogic Server 7.0 SP02 exception was:</p> <pre>&lt;Jul 29, 2003 5:21:55 PM PDT&gt; &lt;Error&gt; &lt;IIOP&gt; &lt;002013&gt; &lt;Complete failure to send exception class java.rmi.MarshalException: java.rmi.MarshalException: IOException while sending; nested exception is: java.io.IOException: Attempt to send message on closed socket. java.rmi.MarshalException: IOException while sending; nested exception is: java.io.IOException: Attempt to send message on closed socket java.io.IOException: Attempt to send message on closed socket at weblogic.iiop.MuxableSocketIIOP.send(MuxableSocketIIOP.java: 362) at weblogic.iiop.EndPointImpl.send(EndPointImpl.java:1078) at weblogic.iiop.OutboundResponseImpl.sendThrowable(OutboundRes ponseImpl.java:194) at weblogic.rmi.internal.BasicServerRef.handleThrowable(BasicSe rverRef.java:473) at weblogic.rmi.internal.BasicServerRef.postInvoke(BasicServerR ef.java:440) at weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServ erRef.java:328) at weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecu teRequest.java:30) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213 ) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189) &gt;</pre> <p>Analysis revealed that close connection messages were not being handled properly. The error was solved with a code change.</p>

<b>Change Request Number</b>	<b>Description</b>
CR124377	<p>WebLogic Server sometimes threw a <code>java.rmi.UnmarshalException</code> when a client application using the thin-client <code>.jar</code> (<code>wlclient.jar</code>) accessed an EJB. On the server, partial exception was:</p> <pre>java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is: java.io.IOException: Serializable readObject method failed internally. java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is: java.io.IOException: Serializable readObject method failed internally at com.ejb_cvps36_EOImpl_WLSkel.invoke(Unknown Source) [...]</pre> <p>On the client, the partial exception was:</p> <pre>java.rmi.MarshalException: CORBA MARSHAL 0 No; nested exception is: org.omg.CORBA.MARSHAL: vmcid: 0x0 minor code: 0 completed: No at com.sun.corba.se.internal.iiop.ShutdownUtilDelegate.mapSystem Exception(ShutdownUtilDelegate.java:97) at javax.rmi.CORBA.Util.mapSystemException(Util.java:65) [...]</pre> <p>This problem did not occur when using <code>weblogic.jar</code> on the client. The code was modified to address this problem.</p>
CR128594	<p>Prior to this Service Pack, WebLogic Server could not handle typecode aliases when reading Java objects on AIX. The code was modified to discard the alias wrapper.</p>
CR128660	<p>WebLogic Server used reflection on inbound contexts that it did not understand. This caused IIOP clients on AIX to fail, because the IBM ORB makes assumptions about the stream format based on the service contexts. The code was fixed to correctly handle service contexts so that IIOP clients can be used on AIX.</p>

# Security

Change Request Number	Description
CR067670	<p>If you configured WebLogic Server to use a custom security realm, and the realm was unavailable, WebLogic Server would not boot. This Service Pack introduces a new startup command, <code>-Dweblogic.security.RealmFailureOk=true</code>, that allows the server to start if the realm is unavailable. The command forces the server to start using the file realm, rather than the configured custom realm.</p> <p>This command is available only for WebLogic Server 6.x, and is not implemented in other server versions.</p> <p><b>Warning:</b> Administrators must be cautious when using this command. If you have a custom realm that is configured only for authorization (not authentication) and the user store is the file realm, booting the server with <code>-Dweblogic.security.RealmFailureOk=true</code> may result in having no security.</p> <p>Use this option only to access the Administration Console and reconfigure the custom realm so that the server can boot normally. Do not allow applications to access the server you have booted with <code>-Dweblogic.security.RealmFailureOk=true</code>.</p>
CR093292	<p>When the CR093292 patch was applied to WebLogic Server Service Pack 5, the server began logging messages such as:</p> <pre data-bbox="327 1016 1163 1317">java.io.IOException: Length is Zero     at weblogic.security.SSL.GenericCipher.input(GenericCipher.java :196)     at weblogic.security.SSL.SSLCiphertext.input(SSLCiphertext.java :68)     at weblogic.security.SSL.SSLSocket.getRecord(SSLSocket.java:114 7) [...]</pre> <p>These messages were intended only for debugging purposes. The code was fixed so that the messages are no longer logged.</p>

Change Request Number	Description
CR093813	<p>The password used to access the encrypted private key in the Node Manager key file, <code>weblogic.nodemanager.keyPassword</code>, was in plain text and accessible.</p> <p>The problem was resolved by the creation of the Node Manager properties file, <code>nodemanager.properties</code>.</p>
CR100703	<p>Authenticated users could become unable to access WebLogic Server 6.1 SP04 after another user had been locked out. This problem could occur with Novell eDirectory. After a user had entered an incorrect password enough times to become locked out, other, authenticated users would become unable to access the server, receiving a 500 error similar to:</p> <pre>&lt;[WebAppServletContext(7814505,security,/security)] Servlet failed with Exception netscape.ldap.LDAPException: error result (53); NDS error: login lockout (-197); DSA is unwilling to perform [...]</pre> <p>The problem was fixed in WebLogic Server by catching the <code>LDAPException</code> thrown in this case, and destroying the connection to the LDAP server.</p>
CR102452	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp</a>.</p>
CR102712	<p>Java clients that used HTTPS tunneling to access EJBs experienced poor performance and increased socket usage, as compared to clients that used HTTP tunneling. WebLogic Server created a new socket, rather than reusing sockets, for each request. This problem was solved with a code fix.</p>
CR105443	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp</a>.</p>
CR105513	<p>This Service Pack includes a new property, <code>weblogic.security.SSL.trustManager</code>, that you can use to specify the custom trust manager if you are using <code>HttpsURLConnection</code> to connect to another server. Using this property enables you to obtain the CA root in the SSL handshake for business logic purposes.</p> <p><b>Warning:</b> This feature is implemented only in WebLogic Server 6.1, and it <i>will not</i> be made available in WebLogic Server 7.x or later releases.</p>

---

Change Request Number	Description
CR108265	<p>In WebLogic Server 6.1 SP04, HTTP requests occasionally failed when <code>CachingRealm.refresh()</code> was called. This servlet synchronizes the cache and the database. The cluster used the RDBMS Realm as the Basic Realm.</p> <p>A simple health check HTTP request calling <code>index.html</code> of an Web application occasionally failed with this exception.</p> <pre data-bbox="327 451 1163 753">&lt;Error&gt; &lt;HTTP&gt; &lt;hanptl02&gt; &lt;managedServer2&gt; &lt;ExecuteThread: '9' for queue: '__weblogic_admin_rmi_queue'&gt; &lt;&gt; &lt;&gt; &lt;101020&gt; &lt;[WebAppServletContext(1014083,healthcheck02,/healthcheck02)] fT_[fuf@fbfg,í Exception ,É,æ,è_, "s,µ,Û,µ,½_B&gt; java.lang.NullPointerException: Start server side stack trace: java.lang.NullPointerException at weblogic.security.acl.GroupImpl.addMember(GroupImpl.java:46) at weblogic.security.acl.OwnerImpl.&lt;init&gt;(OwnerImpl.java:32) at weblogic.security.acl.AclImpl.&lt;init&gt;(AclImpl.java:164) at weblogic.servlet.security.internal.SecurityModule.auditPerm(SecurityModule.java:358)</pre> <p>Analysis revealed that two threads were accessing the <code>CachingRealm</code> simultaneously; for example, the refresh servlet has cleared the cache, but the request to the <code>index.jsp</code> is attempting to find a user that has been cleared from the cache.</p> <p>The problem was solved with a code fix.</p>
CR111041	<p>When a Web Application attempted to make an HTTPS connection and the handshake with the remote server stalled, WebLogic Server never timed out the associated SSL socket. This resulted in execute threads becoming “stuck” indefinitely, eventually locking the server. The code was fixed to ensure that timeout values are enforced with HTTPS connections.</p>
CR112563	<p>WebLogic Server did not encrypt the private key password when storing the password to a file. The code was fixed to automatically encrypt the password when writing the password back to a file. Upon the first booting, WebLogic Server checks to verify that the password is encrypted, and encrypts it in the file if necessary.</p> <p>WebLogic Server uses the encryption service associated with the domain. This means that you can use the password file only with the domain in which it was created.</p> <p><b>Note:</b> Secure a plain text copy of the private key password before you allow WebLogic Server to write the password to a file. You will not be able to retrieve the plain text password from the file after booting the server at this Service Pack level.</p>

<b>Change Request Number</b>	<b>Description</b>
CR113459	<p>In WebLogic Server 6.1 SP05, with CR093813_61sp5.jar, removing the Node Manager properties file caused problems.</p> <p>The Node Manager properties file is always created in the <code>&lt;saved_logs_dir&gt;/NodeManagerInternal</code> directory. This customer periodically archives and deletes the contents of <code>NodeManagerInternal</code>. This removes the Node Manager properties file, so that the certificate password stored in the Node Manager properties file cannot be decrypted, and Node Manager will not work correctly.</p> <p>The problem was solved by a code change to create the Node Manager properties file in the directory specified by <code>user.dir</code>, if the file is not found in <code>NodeManagerInternal</code>, or in the <code>user.dir</code> directory.</p>
CR120850	<p>In WebLogic Server 6.1 SP05, <code>weblogic.net.http.HttpURLConnection</code> did not honor <code>https.nonProxyHosts</code> environment variable. The problem was exhibited in this scenario:</p> <pre>client &lt;--&gt; Proxy &lt;---&gt; Server</pre> <p>The client can be running within WebLogic Server or be a stand-alone Java program. Requests always went through the proxy even if the targeted host was specified in <code>https.nonProxyHosts</code>. If instead, the host is specified in <code>http.nonProxyHosts</code>, the problem does not occur: a direct connection to the host, not to the <code>proxyHost</code> is established, not to the <code>proxyHost</code>, if defined.</p> <p>Analysis revealed that logic for connecting directly to a host specified in <code>https.nonProxyHosts</code>, even when a <code>proxyHost</code> is defined. This problem did not exist for <code>http.nonProxyHosts</code>, only with <code>https.nonProxyHosts</code>.</p> <p>Appropriate logic was developed to connect directly to a host specified in <code>https.nonProxyHosts</code>, even when a <code>proxyHost</code> is defined.</p>
CR121206	<p>In WebLogic Server 6.1 SP05, the number of open LDAP connections kept increasing when using an external LDAP server.</p> <p>The problem was solved with a code fix.</p>
CR121920	<p>In WebLogic Server 6.1 Service Pack 3, calling <code>Socket.getSendBufferSize()</code> when using WebLogic Server JSSE caused the following error:</p> <pre>java.net.SocketException: Socket closed at java.net.PlainSocketImpl.socketGetOption(Native Method) at java.net.PlainSocketImpl.getOption(PlainSocketImpl.java:190) at java.net.Socket.getSendBufferSize(Socket.java:527) [...]</pre> <p>The code was fixed to properly override and delegate <code>getSendBufferSize()</code>.</p>

---

<b>Change Request Number</b>	<b>Description</b>
----------------------------------	--------------------

---

CR121921	The <code>LDAPDelegate.groupContainsInternal()</code> method performed a recursive search of the list of groups without first checking if the list contained the desired group. This performance problem was fixed by updating <code>groupContainsInternal()</code> to first iterate through the group list to determine if it contains the desired group.
----------	--

---

## Servlets

Change Request Number	Description
CR080998	<p>Versions of WebLogic Server 6.1 earlier than SP06 did not support the Range header field definition as described in section 14.35 of the HTTP/1.1 specification. This problem was solved with a code fix.</p>
CR084455	<p>When WebLogic Server 6.1 SP03 was configured to log all HTTP requests in extended format, at the rotation time, the following error is thrown at approximately one minute intervals after the rotation is scheduled (which is the log flush time interval setting):</p> <pre data-bbox="404 581 1233 1227"> ###&lt;Aug 20, 2002 9:00:00 PM PDT&gt; &lt;Error&gt; &lt;HTTP myapp&gt; &lt;host2&gt; &lt;myserver&gt; &lt;ExecuteThread: '6' for queue: 'default'&gt; &lt;system&gt; &lt;&gt; &lt;000000&gt; &lt;Exception flushing HTTP log file&gt; java.io.IOException: Failed to rename log file on attempt to rotate logs at weblogic.servlet.logging.LogManagerHttp.rotateLog(LogManager Http.java:168) at weblogic.servlet.logging.LogManagerHttp.access\$2(LogManagerH ttp.java:148) at weblogic.servlet.logging.LogManagerHttp\$RotateLogTrigger.tri gger(LogManagerHttp.java:432) at weblogic.time.common.internal.ScheduledTrigger.executeLocall y(ScheduledTrigger.java:238) at weblogic.time.common.internal.ScheduledTrigger.execute(Sched uledTrigger.java:229) at weblogic.time.server.ScheduledTrigger.execute(ScheduledTrigg er.java:69) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139 ) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) ###&lt;Aug 20, 2002 9:00:05 PM PDT&gt; &lt;Error&gt; &lt;HTTP myapp&gt; &lt;host2&gt; &lt;myserver&gt; &lt;ExecuteThread: '7' for queue: 'default'&gt; &lt;system&gt; &lt;&gt; &lt;000000&gt; &lt;Exception flushing HTTP log file&gt;... </pre> <p>Analysis revealed that the log had been flushed inappropriately. The problem was solved with a code change to ensure that the log is not flushed on rotation, and to check for a null value for the log file name.</p>
CR086234	<p>The <code>indexDirectories</code> feature of <code>FileServlet</code> was not internationalized, and therefore could not list or use multi-byte file names. This problem was solved with a code fix.</p>

Change Request Number	Description
CR088785	<p>In HTTP access logs, WebLogic Server recorded only the stem portion of the URI, rather than both the stem and query portions, when the <code>cs-uri</code> field was specified. This problem was solved with a code fix.</p>
CR092625	<p>WebLogic Server threw a <code>NullPointerException</code> when you enabled HTTP logging on a Managed Server that was booted with HTTP logging disabled and had no existing log file. On each HTTP access to the Managed Server, the following exception was thrown:</p> <pre data-bbox="327 521 1163 1024"> java.lang.NullPointerException     at weblogic.servlet.logging.LogManagerHttp.log(LogManagerHttp.java:292)     at weblogic.servlet.internal.HttpServer.log(HttpServer.java:865)     at weblogic.servlet.internal.ServletResponseImpl.send(ServletResponseImpl.java:1044)     at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2265)     at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)     at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) </pre> <p>The problem was solved with a code fix.</p>
CR095945	<p>In WebLogic Server 6.1 SP05 running under Unix, <code>CGIServlet</code>, which extracts the <code>cgiscripts</code> in a WAR so that it can execute them, was calling the scripts without setting the current working directory.</p> <p>A fix was implemented to ensure that the current working directory is set so that <code>cgi</code> scripts can call subscripts, even in WAR webapps.</p>
CR095981	<p>In WebLogic Server 6.1 SP01, SP02, SP03, and SP04, <code>&lt;charset-mapping&gt;</code> in <code>weblogic.xml</code> is ignored when compiling JSP file with <code>precompile=true</code>.</p> <p>As a result, some of double bytes characters are garbled because of the mismatch between the character's encoding specified in <code>&lt;charset-mapping&gt;</code> and in the compiled classes.</p> <p>Analysis revealed an error in the precompiler. The problem was solved with a code fix.</p>

Change Request Number	Description
CR096459	<p>When invoking a flush in a Servlet or JSP in HTTP 1.0, WebLogic Server sometimes failed to close the socket, causing the client to wait for a period of time until the socket timed out. The problem was solved with a code fix.</p>
CR100590	<p>An update to the <code>HttpClusterServlet</code> implementation changed the default SSL port number in the <code>WebLogicCluster</code> parameter to 443. The updated implementation also failed to report an error when the <code>WebLogicCluster</code> parameter was not specified correctly.</p> <p>A code fix was made to ensure that the <code>HttpClusterServlet</code> implementation matched the earlier behavior:</p> <ul style="list-style-type: none"><li>■ <code>WebLogicCluster</code> again uses the format <code>host:port:sslport</code> for specifying the SSL port number.</li><li>■ WebLogic Server logs an error if the SSL port is not specified, and uses the default SSL port of 7002.</li></ul>
CR100645	<p>Objects that implement the <code>HttpSessionAttributeListener</code> interface did not receive the correct value when calling <code>HttpSessionBindingEvent.getValue()</code> on events received via <code>attributeReplaced()</code>. Instead of returning the old attribute value, as stated in the Servlet 2.3 specification, <code>getValue()</code> returned the newer value. The problem was solved with a code fix.</p>
CR101061	<p>Prior to this Service Pack, a call to <code>weblogic.version</code> did not show the correct version information if a patch had been applied to the WebLogic Server instance. The code was modified so that <code>weblogic.version</code> now shows the correct version and patch information in the format:</p> <pre>WebLogic Server version date build with patch [, patch] [...]</pre> <p>For example:</p> <pre>WebLogic Server 8.1 03/20/2003 246620 with CRXXXX, CRXXXX, CRXXXXX</pre>

---

Change Request Number	Description
-----------------------	-------------

---

CR101838	<p>In WebLogic Server SP04, there was a change in the behavior of the <code>CGIServlet</code> that could affect the execution of CGI scripts having no filename extension. For example, if a <code>web.xml</code> file defined the CGI directory and Servlet mapping in the following way:</p> <pre>&lt;param-name&gt;cgiDir&lt;/param-name&gt; &lt;param-value&gt;/home/user/cgi-bin&lt;/param-value&gt; &lt;/init-param&gt; &lt;/servlet&gt; &lt;servlet-mapping&gt;   &lt;servlet-name&gt;CGIServlet&lt;/servlet-name&gt;   &lt;url-pattern&gt;/cgi-bin/*&lt;/url-pattern&gt; &lt;/servlet-mapping&gt;</pre> <p>and stored a script named <code>myscript</code> in the <code>/home/user/cgi-bin</code> directory, a URL such as <code>http://localhost/mywebapp/cgi-bin/myscript</code> would map to <code>/home/user/cgi-bin</code> without specifying the script name. (This problem did not occur with scripts having an extension, such as <code>myscript.ksh</code>).</p> <p>The code was fixed to make the behavior match earlier releases of <code>CGIServlet</code>. Using the above example, the code fix ensures that the URL <code>http://localhost/mywebapp/cgi-bin/myscript</code> maps to <code>/home/user/cgi-bin/myscript</code>.</p>
----------	--

---

<b>Change Request Number</b>	<b>Description</b>
CR102262	<p>After parsing an HTTP request containing a null HTTP-Version field, WebLogic Server would throw the following exception:</p> <pre>java.lang.NullPointerException     at weblogic.servlet.internal.ServletRequestImpl.initInputEncoding(ServletRequestImpl.java:727)     at weblogic.servlet.internal.ServletRequestImpl.mergePostParams(ServletRequestImpl.java:565)     at weblogic.servlet.internal.ServletRequestImpl.parseQueryParams(ServletRequestImpl.java:489)     at weblogic.servlet.internal.ServletRequestImpl.getParameter(ServletRequestImpl.java:686)     at weblogic.servlet.internal.ServletRequestImpl.initSessionInfo(ServletRequestImpl.java:1481)     at weblogic.servlet.internal.ServletRequestImpl.getSession(ServletRequestImpl.java:1345)     at weblogic.servlet.internal.ServletContextImpl.invokeServlet(ServletContextImpl.java:1010)     at weblogic.servlet.internal.ServletContextImpl.invokeServlet(ServletContextImpl.java:910)     at weblogic.servlet.internal.ServletContextManager.invokeServlet(ServletContextManager.java:279)     at weblogic.socket.MuxableSocketHTTP.invokeServlet(MuxableSocketHTTP.java:403)     at weblogic.socket.MuxableSocketHTTP.execute(MuxableSocketHTTP.java:285)     at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:130)</pre> <p>The code was fixed to use HTTP/0.9 as the default version when none is specified in the request.</p>

---

---

Change Request Number	Description
CR102574	Forcibly shutting down a client to a servlet caused increased memory usage in WebLogic Server, potentially leading to an <code>OutOfMemoryError</code> . The problem was solved with a code fix.
CR102689	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEAO3_36.00.jsp</a> .
CR102769	<p>When logging HTTP transactions using extended log format, WebLogic Server did not record query parameters (<code>cs-uri-query</code>) for requests made from one Servlet or JSP to another. Also, WebLogic Server recorded the URI (<code>cs-uri-stem</code>) of the forwarded request, rather than the original request URI from the client.</p> <p>The code was fixed to ensure that the URI and query string of the original request are recorded in <code>access.log</code> when logging forwarded HTTP transactions.</p>
CR103059	<p>A <code>&lt;url-pattern&gt;</code> value with a single character in a <code>&lt;servlet-mapping&gt;</code> element was not honored in a <code>web.xml</code> deployment descriptor. For example, <code>&lt;url-pattern&gt;*.f&lt;/url-pattern&gt;</code> produced a 404 File Not Found error when attempting to access <code>welcome.f</code>, but <code>&lt;url-pattern&gt;*.oop&lt;/url-pattern&gt;</code> worked properly when attempting to access <code>welcome.oop</code>.</p> <p>This problem has been resolved so that single character <code>&lt;url-patterns&gt;</code> now work as expected.</p>
CR103256	A code change resulted in performance improvement in JDBC regarding bubble caches in <code>JDBCSessionContext</code> and <code>JDBCSessionData</code> .
CR103289	<p>The <code>HttpClusterServlet</code> was not correctly parsing the session id from post data. If you sent a request through <code>HttpClusterServlet</code> to a cluster, establishing a session, and then sent a second request without a cookie but with the session id in the post parameters, the servlet did not recognize the session.</p> <p>The servlet is now able to extract the session id from the post data.</p>
CR103339	Servlet output capitalized the <code>Transfer-Encode</code> value, <code>Chunked</code> , instead of using lowercase <code>chunked</code> as required by the HTTP/1.1 specification. The code was fixed to ensure that Servlets use the <code>Transfer-Encode</code> type of “chunked” in lowercase.
CR103925	The <code>setAttribute</code> method only checked for <code>hashCode</code> equality. It now also checks the result of the <code>equals</code> method for the old and new objects.
CR104975	If a log rotation failed, WebLogic Server failed to reopen the log file. When this occurred, the trigger to flush the log file could throw a <code>java.io.IOException: Bad file descriptor</code> . The problem was solved with a code fix.

---

<b>Change Request Number</b>	<b>Description</b>
CR105016	<p>HttpClusterServlet failed to increase its failure count if a WebLogic Server in the cluster was hung. This caused HttpClusterServlet to go into an infinite loop if each server in a cluster slept for a time longer than HungServerRecoverSeconds, or if 2 out of 3 servers in a cluster slept longer than HungServerRecoverSeconds. The code was fixed to ensure that the failure count is properly incremented.</p>
CR105339	<p>In Service Pack 5, WebLogic Server did not create variable definitions for tag calls that used the TagExtraInfo class. The code was fixed to ensure that the variable definitions are again created (as they were in Service Pack 4 and earlier).</p>
CR106186	<p>If you set the buffer size to zero in a servlet, it would fail to respond and would initiate an infinite loop in the server. WebLogic Server would ultimately display a warning similar to:</p> <pre>"Suspend Checker Thread" prio=10 tid=0x23eb90 nid=0xfec runnable &lt;May 14, 2003 10:53:34 AM PDT&gt; &lt;Warning&gt; &lt;WebLogicServer&gt; &lt;000337&gt; &lt;ExecuteThread: '10' for queue: 'default' has been busy for "1,181" seconds working on the request "Http Request: servlet_uri", which is more than the configured time (StuckThreadMaxTime) of "600" seconds.&gt;</pre> <p>This problem was solved with a code fix.</p>
CR107419	<p>JSP code that intended to set double byte characters as a parameter's value—for example:</p> <pre>&lt;jsp:include page="included.jsp"&gt; &lt;jsp:param name="title" value="&lt;%= java.net.URLEncoder.encode(\"[double byte characters]\") %&gt;"/&gt; &lt;/jsp:include&gt;</pre> <p>—resulted in specified double byte characters being changed to their URL-encoded code. This problem has been resolved.</p>
CR108607	<p>WebLogic Server produced errors when applications used XSL with the FOP output method to get an image from an archive file. This problem did not occur for applications deployed in exploded format. This problem was solved with a code fix.</p>
CR108350	<p>The Administration Console incorrectly indicated that the log file format could be dynamically changed between Common Log Format (CLF) and Extended Log Format (ELF), when such a change actually requires a server reboot. The code was changed to properly indicate the required server reboot when changing this configuration parameter.</p>

Change Request Number	Description
CR109958	<p>When processing requests through a proxy servlet, WebLogic Server only honored the <code>SecureProxy</code> setting for incoming requests that used HTTPS. If the incoming request used HTTP, WebLogic Server did not use an HTTPS connection even when <code>SecureProxy</code> was enabled in the proxy servlet. This problem was solved with a code fix.</p>
CR110798	<p>Calling <code>weblogic.servlet.security.ServletAuthentication.killCookie(req)</code> in a JSP caused the session to remain in WebLogic Server without ever being cleaned up. The code was fixed to ensure that the session is invalidated just before <code>killCookie(req)</code> completes.</p>
CR110914	<p>If <code>TrackingEnabled</code> was set to false, WebLogic Server created a new session for each request, but the sessions were not getting invalidated.</p> <p>The code was modified to invalidate a session immediately if <code>TrackingEnabled</code> is set to false. This is the correct behavior.</p>
CR111752	<p>Under certain conditions, WebLogic Server threw a <code>NullPointerException</code> when using <code>CGIServlet</code> with the <code>useByteStream</code> parameter set to true. The problem occurred when using framesets where one frame contained static URL links and another frame used <code>CGIServlet</code>. If a user selected the frame containing static links before the other frame completed downloading a page, an IO exception was caught and presented to the user as:</p> <pre>java.lang.NullPointerException     at weblogic.utils.Executable\$Drainer.run(Executable.java:366)</pre> <p>This problem was solved with a code fix.</p>
CR112799	<p>WebLogic Server attempted to write to an output stream even after an <code>IOException</code> occurred. This led to 100% CPU utilization if an unexpected socket disconnection occurred with a Web Application that did not handle <code>IOException</code>.</p> <p><code>org.apache.xml.serialize.XMLSerializer</code> ignores <code>IOException</code> until the end of its process. This caused the problem to occur if an <code>IOException</code> was thrown in the middle of returning XML documents as part of an HTTP response.</p> <p>The code was modified to ensure that writes to an output stream stop after an <code>IOException</code> occurs.</p>

<b>Change Request Number</b>	<b>Description</b>
CR120440	<p>When multiple Web Applications were deployed in a Single Sign-On configuration and one application called <code>weblogic.servlet.security.ServletAuthentication.invalidateAll(request)</code>, the <code>HttpSessionListeners</code> in the other applications were not invoked until their session timeouts occurred. This happened because only the session associated with the first Web Applications was registered for invalidation; after the user was authenticated, subsequent sessions were not registered.</p> <p>The code was fixed to ensure that both the session ID and context path of all Web Applications are registered for invalidation as necessary by <code>invalidateAll(request)</code>.</p>
CR122177	<p>The fix for CR060023 in Service Pack 3 caused the <code>FileServlet</code> to return a response code of 200 instead of 404 when a file is not found. The code was fixed to return 404 for when a file is not found.</p>
CR121846	<p>In WebLogic Server Service Pack 3, it was possible for the server to write standard log entries to a log file before the writing Extended Log Format headers. This situation could occur during a log rotation when multiple threads attempted to write to the new log file at the same time.</p> <p>The code was fixed to ensure that the thread handling the log rotation has exclusive access to the new log file until after the log headers are written.</p>

Change Request Number	Description
CR125718	<p>WebLogic Server Service Pack 5 could throw a <code>NullPointerException</code> when used with Apache and Netegrity's SiteMinder. An initial request forwarded through Apache to SiteMinder, and authenticated by SiteMinder, would also be successfully authenticated by WebLogic Server. However, if the user used the browser's Back button to return to the login page and authenticated again using a different username and password, WebLogic Server threw a <code>NullPointerException</code>:</p> <pre>java.lang.NullPointerException     at weblogic.servlet.security.internal.SecurityModule.logoutSession(SecurityModule.java:386)     at weblogic.servlet.security.internal.SecurityModule.checkAuthenticate(SecurityModule.java:292)     at weblogic.servlet.security.ServletAuthentication weak(ServletAuthentication.java:353)     at com.uprr.security.weblogic.SiteMinderAuthFilter.doPreAuth(Unknown Source)     at weblogic.servlet.security.AuthFilter.service(AuthFilter.java:51)     at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:262)     at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:198)     at weblogic.servlet.internal.RequestDispatcherImpl.include(RequestDispatcherImpl.java:530)     at weblogic.servlet.internal.RequestDispatcherImpl.include(RequestDispatcherImpl.java:350)     at weblogic.servlet.security.internal.ServletSecurityManager.checkAccess(ServletSecurityManager.java:144) [...]</pre> <p>This problem was solved with a code fix.</p>

---

# SNMP

---

Change Request Number	Description
CR088000	<p>When WebLogic Server 6.1 SP04, a SNMP shutdown trap was not received.</p> <pre>8-Oct-02 15:57:24 FAILURE mytest!0[41]!oam/systest/snmp/oam_snmp!2[69]!if!3[86]!weblog ic.qa.tests.oam.systest.snmp.trap.SNMPTrapTest.testServerShu tdownTrap() wlsServerShutdownTrap wasn't received. PARAMETERS: managedServerName = oamserver1 adminServerName = adminServer</pre> <p>Analysis revealed that SNMP traps were not generated for Managed Server if the listen address or listen port are overridden with command line option.</p> <p>The problem was resolved with a logic change to use the listen port value from ServerMBean, rather than the one in the URL.</p>

---

# Web Services

---

Change Request Number	Description
CR099255	<p>When using two-way SSL to invoke a Web Service, and the client code uses the WSDL of the Web Service in its invocation, the client API now correctly sends the certificates both when making the WSDL lookup <i>and</i> when invoking the operation. Previously, certificates were sent only for the WSDL lookup and not for the actual invocation of the operation.</p> <p>This problem was solved with a code fix.</p>
CR099534	<p>The <code>wsgen</code> Ant task has been optimized to use a new classloader for every Web Service it assembles from a single <code>build.xml</code> file.</p> <p>Previously, <code>wsgen</code> would use a single classloader for its entire execution, even if there were multiple Web Services listed in the <code>build.xml</code> file, and all auxiliary classes for each subsequent Web Service would be obtained from the single classloader and then copied, which was a sub-optimal way of generating all the needed class files. Depending on the number of Web Services in the <code>build.xml</code> file that had to be assembled, a full execution of <code>wsgen</code> could take over 30 minutes.</p> <p>This problem was solved with a code fix.</p>

---

<b>Change Request Number</b>	<b>Description</b>
CR101383	<p>WebLogic Web Services now correctly represent a null <code>xsd:dateTime</code> value in the SOAP response as <code>xsi:nil='true'</code>.</p> <p>Previously, if a WebLogic Web Service returned an <code>xsd:dateTime</code> value that was null, it incorrectly represented it in the SOAP response as <code>xsi:null='1'</code>, which is how the 1999 version of XML Schema defined null values. WebLogic Web Services support the 2001 version of XML Schema, which defines null values as <code>xsi:nil='true'</code>.</p> <p>This problem was solved with a code fix.</p>
CR102677	<p>When using the XML Registry for external entity resolution, WebLogic Server no longer returns an error when resolving DTDs using the Public ID.</p> <p>This problem was solved with a code fix.</p>
CR106892	<p>The generated WebLogic Web Services client stubs, when sending an XML document to a Web service as an <code>org.w3c.dom.Element</code> object, now correctly preserve any new-lines characters inside the body of an element in the XML file. Previously the client stubs would incorrectly convert these new-line characters to spaces.</p> <p>This problem was solved with a code fix.</p>
CR111151	<p>WebLogic Web Services now correctly send the utf-8 encoding header in SOAP Fault responses that are generated as a result of an exception in the invocation of the Web Service. The complete correct header is:</p> <pre>&lt;?xml version="1.0" encoding="utf-8"?&gt;</pre> <p>Previously, WebLogic Web Services did not include this header in SOAP Fault responses.</p> <p>This problem was solved with a code fix.</p>

## WTC-ATMI

---

<b>Change Request Number</b>	<b>Description</b>
CR107323	<p>In WebLogic Server 6.1 SP03 configured to use Tuxedo 8.0, when WebLogic Server shut down abnormally (CTRL-C) while there were uncommitted Tuxedo transactions, upon restart, the WebLogic Server instance hung. The problem was exhibited in this scenario:</p> <ol style="list-style-type: none"><li>1. WebLogic was killed while there are still uncommitted transactions with Tuxedo</li><li>2. WTC is configure to be started "ON_DEMAND"</li><li>3. Start WebLogic</li><li>4. WebLogic hangs with the reported problem</li></ol> <p>The problem was not exhibited when:</p> <ol style="list-style-type: none"><li>1. WebLogic was killed while there are still uncommitted transactions with Tuxedo</li><li>2. WTC is configure to be started "ON_DEMAND"</li><li>3. Remove WebLogic tlog files "/*.tlog"</li><li>4. Start WebLogic</li><li>5. WebLogic starts without any problem</li></ol> <p>or when:</p> <ol style="list-style-type: none"><li>1. WebLogic was killed while there are still uncommitted transactions with Tuxedo</li><li>2. WTC is configure to be started "ON_STARTUP"</li><li>3. Start WebLogic</li><li>4. WebLogic starts without any problem</li></ol> <p>When the hang occurs after a restart, all the threads in the default Group hang with the following stack trace:</p> <pre>at java.lang.Object.wait(Native Method) at java.lang.Object.wait(Object.java:415) at weblogic.wtc.jatmi.TuxXidRply.get_specific_reply (TuxXidRply.java:203)at weblogic.wtc.gwt.TuxedoXA.internalCommit(TuxedoXA.java:259) at weblogic.wtc.gwt.TuxedoXA.commit(TuxedoXA.java:328) at weblogic.wtc.gwt.OatmialCommitter.execute(WTCStartup.java:27 67)at...</pre> <p>The problem was solved by modifying the <code>weblogic.wtc.gwt.WTCStartup</code> method to sleep for one second after each <code>Kernel.execute(myCommitter)</code>.</p>

---

---

<b>Change Request Number</b>	<b>Description</b>
----------------------------------	--------------------

---

CR120681	In WebLogic Server 6.1 SP02, WebLogic Tuxedo Connector did not properly translate local resource name to remote service name. A correction to the translation code solved the problem.
----------	---

---

## XML

---

<b>Change Request Number</b>	<b>Description</b>
CR102397	<p>The Xerces parser always attempted a call to <code>System.getProperty()</code>. This would throw an exception in an Applet, which does not have permission to retrieve properties. The following partial exception could be observed in an Applet making a JMS call:</p> <pre>----- Linked Exception ----- java.rmi.MarshalException: failed to marshal connectionCreate(Lweblogic.jms.dispatcher.DispatcherWrapper; ); nested exception is: java.rmi.UnexpectedException: Failed to parse descriptor file; nested exception is: java.security.AccessControlException: access denied (java.util.PropertyPermission weblogic.apache.xerces.maxentityrefs read) [...]</pre> <p>The code was modified so that the parser does not attempt to read system properties with Applet clients.</p>

---

# WebLogic Server 6.1 Service Pack 5 Solutions

The following sections describe problems resolved in WebLogic Server 6.1 Service Pack 5.

- Core
- Connector
- Console
- Core
- Deployment

- EJB
- Installer
- JDBC
- jDriver
- JMS
- JNDI
- JSP
- JTA
- Miscellaneous
- Plug-ins
- RMI
- RMI-IIOP
- Security
- Servlets
- WLS-Tour/Examples
- Web Services
- WebLogic Tuxedo
- XML

---

# Cluster

---

<b>Change Request Number</b>	<b>Description</b>
CR082443	<p>In an environment with multiple clusters using the same multicast address, large log files resulted, causing overly frequent file rollover.</p> <p>The debug messages are similar to:</p> <pre>###&lt;Jul 23, 2002 4:06:26 PM EDT&gt; &lt;Debug&gt; &lt;Cluster&gt; &lt;ustrntda01&gt; &lt;wts-cluster_test&gt; &lt;ExecuteThread: '9' for queue: 'default'&gt; &lt;&gt; &lt;&gt; &lt;000000&gt; &lt;dropped fragment from foreign domain/cluster domainhash=95475927 clusterhash=-1674453961&gt;</pre> <p>The problem occurred because, in <code>weblogic.cluster.ClusterDebug.java</code>, the flag controlling whether or not debug messages are logged was set to true. The problem was solved with a code fix.</p>
CR087240	<p>HTTP session failover resulted in a <code>ClassCastException</code>, when the Managed Server hosting the session was shut down. Failover was successful when error message was ignored.</p> <p>This problem was reliably reproduced in this configuration:</p> <ol style="list-style-type: none"><li>1) Two clustered server instances, running 6.1 SP02.</li><li>2) Cluster hosted a WLP 4.0 sSP002 portal application.</li><li>3) One IPlanet web server with WebLogic Server plug-in pointing to the two-node cluster.</li></ol> <p>The problem was not consistently reproducible in other configurations.</p> <p>This problem was associated with a problem in the WebLogic Server shutdown sequence.</p> <p>This problem was resolved by a new service to shutdown <code>RMIServerService</code>.</p>

---

CR092146

Failover did not work for a stateful session bean in a cluster. The request URL contained the local server name instead of the cluster DNS name.

The following exception occurs:

```
java.rmi.NoSuchObjectException: Unable to locate EJBHome:
'statefullCluster' on server: 't3://172.23.135.83:7600 at
weblogic.ejb20.internal.HomeHandleImpl.getEJBHome(HomeHandle
Impl.java:80) at
weblogic.ejb20.internal.HandleImpl.getEJBObject(HandleImpl.j
ava:204) at
com.bea.cluster.testClusterEJBServlet.doGet(testClusterEJBSe
rvlet.java:137) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:740)
at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:262) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:198) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet
(WebAppServletContext.java:2637) at
weblogic.servlet.internal.ServletRequestImpl.execute(Servlet
RequestImpl.java:2359) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

Problem was solved by modifying the `getURL()` method in

```
weblogic.ejb20.internal.BaseEJBHome.java
```

to provide the cluster address as the host portion for the URL.

---

- 
- CR093809      A stack trace resulted from a error looking up a session. A secondary server instance was being shutdown. While making a log entry for a request it served, the HTTP server tried to get the user information from the session. As a result it tried to look up the secondary server even though it was down, resulting in this stack trace.
- ```
<Dec 26, 2002 7:17:55 PM EST> <Error> <HTTP Session> <Error
looking up session for
id:2LcR6Oo12IC5qNtFZhThJcLYVYEXt9KWG2ciXmAXZ4XiBAHPiHx4!-137
9505194!stcasfi01b.usa-ed.net!8001!7002!-1127797657!stcasfi0
2b.usa-ed.net!8001!7002 java.rmi.ConnectIOException: Server
is being shut down Start server side stack trace:
java.rmi.ConnectIOException: Server is being shut down at
weblogic.rjvm.RJVMImpl.dispatchRequest(RJVMImpl.java:692) at
weblogic.rjvm.RJVMImpl.dispatch(RJVMImpl.java:666) at...
```
- The primary server correctly selected a new secondary from the available server list, and no session data was lost.
- This problem was resolved with a code fix. Now, the server obtains user information from `ThreadLocal`, instead of the session.
- 
- CR094561      Please review the security advisory information at <http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-26.01.jsp>.
- 
- CR101609      In WebLogic Server 6.1 SP04, session replication stopped working after a failure to replicate a non-serialized object. Invocation of a JSP that tried to set non-serialized data into the session object, resulted in the expected exception:
- ```
<Mar 19, 2003 2:58:23 PM PST> <Error> <Cluster> <All session
objects should be serializable to replicate. Please check the
objects in your session. Failed to replicate non serializable
object>
```
- Then, for all subsequent requests, sessions were not replicated, and this exception occurred:
- ```
<Mar 19, 2003 2:58:48 PM PST> <Debug> <Cluster> <Unable to
create secondary for -8956818414963087828>
```
- The problem was solved with a code change. Now, when a non-serializable object is encountered, the method returns, instead of trying other secondaries.
- 
- CR102655      Duplicate of CR101609. See above.
-

# Connector

| Change Request Number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR086251              | <p>In WebLogic Server 6.1 SP03 the Messaging Bridge Adapter threw a <code>NullPointerException</code> in <code>unregisterResource</code> during shutdown. This occurred if messages have been sent but not consumed before the shutdown of the server has been initiated. The problem did not occur if Messaging Bridge is not deployed and the Adapter is.</p> <p>The configuration was: <code>jms-xa-adp.rar</code> Messaging Bridge Adapter and a Messaging Bridge between MQSeries 5.2 and JMS</p> <p>The problem was corrected with a code fix to check <code>wrappedXARes</code> before attempting to <code>unregister</code>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| CR090002              | <p>In WebLogic Server 6.1 SP03, <code>JCA-Connection.close</code> throws <code>java.lang.reflect.UndeclaredThrowableException</code> instead of <code>ResourceException</code></p> <p>A code fix was implemented to ensure that the resource adaptor's original exception is passed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| CR090792              | <p>In WebLogic Server 6.1 SP03 and SP04, RAR deployment failed after any of its descriptor parameters were changed in console.</p> <p>Analysis revealed that any descriptor change in console caused reset of pool parameters if they had not been changed from default values. Pool parameters were inappropriately defined:</p> <pre>&lt;pool-params&gt;   &lt;initial-capacity&gt;0&lt;/initial-capacity&gt;   &lt;max-capacity&gt;0&lt;/max-capacity&gt;   &lt;capacity-increment&gt;0&lt;/capacity-increment&gt;   &lt;shrinking-enabled&gt;false&lt;/shrinking-enabled&gt;   &lt;shrink-period-minutes&gt;0&lt;/shrink-period-minutes&gt; &lt;/pool-params&gt;</pre> <p>This caused deployment failure due to max capacity value 0 which is prohibited. No activities could be performed on RA after that.</p> <p>In 6.1, <code>ResourceAdapterComponentMBean</code> was used when getting descriptor values. However, <code>ResourceAdapterComponentMBean</code> constructor did not automatically have defaults set (the way <code>ConnectorComponentMBean</code> does) and also did not set defaults manually. The fix was to set them manually.</p> |

## Console

| Change Request Number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR062102              | <p>A customer with large application (40 MB) migrating from 6.1 SP01 experienced network overload and slow WebLogic Server startup, as a result of WebLogic Server copying the application to Managed Servers at startup.</p> <p>A code change was made to address this problem. Now, unchanged applications are not copying to Managed Servers at startup.</p> <p>Use the <code>forceApplicationCopy</code> parameter to cause applications to be copied to Managed Servers at startup, whether changed or unchanged. See “New Parameter for Forced Application Update” on page 1-4.</p>                                                                                                                                                                                                                                                                                                                                                  |
| CR069284              | <p>When the "Auto Deployment Enabled" attribute on the Domain -&gt; Configuration -&gt; Applications page was turned off, application components were still deployed automatically when application file was copied to the applications directory.</p> <p>Investigation revealed that the “Auto Deployment Enabled” attribute is not used in WebLogic Server. Instead, auto deployment is controlled by the <code>DomainMBean.setProductionModeEnabled</code> property, which is set at startup on the command line.</p> <p>The <code>AutoDeployedEnabled</code> attribute has been removed from the Administration Console.</p>                                                                                                                                                                                                                                                                                                           |
| CR084607              | <p>In WebLogic Server 6.1 SP03, an <code>InstanceNotFoundException</code> occurred when creating a JMS Topic at runtime in a cluster. The Topic was created with <code>JMSHelper.createPermanentTopicAsync</code> in a startup class targeted to a Managed Server in a cluster. After the Topic was created, when the user clicked Destination Link for the JMS Server in the Administration Console, this message was displayed :</p> <pre data-bbox="400 1133 1233 1438"> java.lang.reflect.UndeclaredThrowableException: javax.management.InstanceNotFoundException: mydomain:Name=testTopic12,Type=JMSTopic at com.sun.management.jmx.MBeanServerImpl.getMBean(MBeanServerImpl.java:1680) at com.sun.management.jmx.MBeanServerImpl.getAttribute(MBeanServerImpl.java:1152) at weblogic.management.internal.MBeanProxy.getAttribute(MBeanProxy.java:254) at weblogic.management.internal.MBeanProxy.invoke(MBeanProxy.java:187) </pre> |

CR088842 In WebLogic Server 6.1 SP02, in a configuration with an Administration Server and three Managed Servers, a deadlock when one of the Managed Servers was restarted. Analysis revealed that `lookupServerRuntime()` on a Managed Server resulted in unnecessary `getMBean()` calls to the Administration Server. The problem was resolved with a code fix.

---

CR089747 In WebLogic Server 6.1 SP03, on Solaris Sparc 2.8, the domain log file did not rotate by time. Under Windows 2000, 3 log files were created, then rotation stopped. When old log files were deleted, rotation started again. For Windows 2000, log rotation worked after an upgrade from SP01 to SP03, but not with the fresh installation of SP03. The configuration for the domain log is:

```
<Log FileName="config/mydomain/logs/wl-domain.log"
FileTimeSpan="1" Name="mydomain" RotationType="byTime"
RotationTime=10:00/>
```

The problem was solved with a code fix.

---

CR091359 In WebLogic Server 6.1 SP03 and SP04, the Administration Console option: Web Application->*webapp1*->Monitoring->Monitor all Active Web Applications... did not work for a web application inside an .ear file. The problem was resolved with a code fix.

---

CR093409 A configuration with one Administration Server and five Managed Servers, running under WebLogic Server 6.1 SP02, Solaris 2.8, and JDK1.3.1\_04, an attempt to deploy an EAR with 3 EJB components using `weblogic.deploy` utility from a different machine caused the Administration Server to hang. The thread dump reported a deadlock. The stack trace contained:

```
Execute thread 4(admin_rmi queue) holds lock of the
com.sun.management.jmx.MBeanServerImpl and waiting to acquire
lock on weblogic.logging.LogManager. Execute thread 8(default
queue) holds lock of the weblogic.logging.LogManager and
waiting to acquire lock on
com.sun.management.jmx.MBeanServerImpl.
```

Analysis revealed a deadlock in the `LogManager` and `FileStreamLogger`. A code fix resolved the error.

---

- 
- CR096726      In WebLogic Server 6.1 SP02, invocation of `MBeanHome.deleteMBean()` resulted in a null point exception.
- The configuration was four managed servers, each on a different physical machine, with a JMX client running against each (on one-to-one basis). The JMX clients encounter `NullPointerException` when doing a `MBeanHome.deleteMBean`.
- Analysis revealed that the problem was related to the asynchronous deletion of multiple Managed Servers. One Managed Servers deleted its references in other mbeans before deleting itself, while another Managed Servers does the same thing, resulting in null pointer exception.
- The problem was resolved by implementing `try/except` blocks around the code that obtains the metadata for an MBean (i.e., `mbean.getMBeanInfo()`). When deleting an MBean, a list of current MBeans in the `MBeanServer` is first obtained, then references to the bean being deleted are removed. The problem is, is that other delete threads are doing the same thing and thus may delete a bean in our (static) list, so when we go to get the meta-data on the bean to update it, we NPE. This patch ignores that NPE and we go to the next one in our list.
- 
- CR096950      Please review the security advisory information at <http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.jsp>.
- 
- CR098623      When a portal application, such as `<beahome>\wlportal4.0\applications\portal` was deployed as an EAR file, it was not possible to modify parameters in `application-config.xml` with the Administration Console. A code fix resolved the problem.
-

CR098739

In WebLogic Server 6.1 SP02, a Java deadlock was detected in an Administration Server. The thread dump indicated a java level deadlock as follows, where Thread 8 holds the lock on RemoteMBeanServerImpl and tries to obtain one on LogManager, which is held by Thread 6 who is in turn waiting on RemoteMBeanServerImpl.

```
Java Stack for "ExecuteThread: '8' for queue: '__weblogic_admin_rmi_queue'":
at weblogic.logging.LogManager.addSubsystem(LogManager.java:
58) - waiting to lock <f55b6358> (a weblogic.logging.Log
Manager) at weblogic.logging.LogOutputStream.getLogManager
(LogOutputStream.java:32) at weblogic.logging.LogOutput
Stream.error(LogOutputStream.java:63) at weblogic.management.
internal.Helper.error(Helper.java:1391) at weblogic.
management.internal.Helper.error(Helper.java:1404) at
weblogic.management.internal.ConfigurationMBeanImpl.
postDeregister(ConfigurationMBeanImpl.java:900) at com.sun.
management.jmx.MBeanServerImpl.postDeregisterInvoker
(MBeanServerImpl.java:2314) at com.sun.management.jmx.MBean
ServerImpl.unregisterMBean(MBeanServerImpl.java:967) - locked
<f550eec8> (a weblogic.management.internal.RemoteMBean
ServerImpl) at weblogic.management.internal.RemoteMBeanServer
Impl.unregisterMBean(RemoteMBeanServerImpl.java:213) at ..
```

Java Stack for "ExecuteThread: '6' for queue: 'default'":

```
at com.sun.management.jmx.MBeanServerImpl.getMBean(MBean
ServerImpl.java:1672) at com.sun.management.jmx.MBeanServer
Impl.getAttribute(MBeanServerImpl.java:1150) at weblogic.
management.internal.MBeanProxy.getAttribute(MBeanProxy.java:
254) at weblogic.management.internal.MBeanProxy.invoke
(MBeanProxy.java:187) at $Proxy7.isStdoutEnabled(Unknown
Source) at weblogic.logging.ConsoleLogger.isLog(Console
Logger.java:81) at weblogic.logging.ConsoleLogger.log
(ConsoleLogger.java:70) at
weblogic.logging.LogManager.log(Log
Manager.java:260) - locked <f55b6358> (a
weblogic.logging.LogManager) at
weblogic.logging.MessageLogger.log(MessageLogger.java:17) at
weblogic.t3.srvr.T3SrvrLogger.logRemovingClientContextSoftDi
sconnect(T3SrvrLogger.java:1204) at
weblogic.t3.srvr.ClientContext.dieIfTimedOut(ClientContext.j
ava:512) at ...
```

The problem was solved by a code change to remove unnecessary locks in LogManager.

---

## Core

---

| <b>Change Request Number</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR036602                     | <p>The <code>ClusterMBean.setClusterAddress()</code> method accepted an illegal value and did not throw an <code>IllegalArgumentException</code>.</p> <p>The problem was resolved by checking that the cluster address is during server startup.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR036603                     | <p>The <code>ClusterMBean.setMultiCustAddress()</code> method accepted an illegal value and did not throw an <code>IllegalArgumentException</code>.</p> <p>The problem was resolved by a code change.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CR077170                     | <p>In WebLogic Server 6.1 SP03, stopping a Managed Server after its Administration Server was shutdown caused an exception. The problem occurred with this sequence of actions:</p> <ol style="list-style-type: none"><li>1. Start an Administration Server and a Managed Server.</li><li>2. Shutdown the Administration Server.</li><li>3. Invoke the 'stop' method on the <code>ServerConfigMBean</code> of the Managed Server.</li></ol> <p>This exception resulted:</p> <pre>Unexpected Exception Start server side stack trace:<br/>java.rmi.ConnectException: Unable to get direct or routed<br/>connection to:<br/>'-19546889165621794S:dwarkamai:[7001,7001,-1,-1,7001,-1,-1]:<br/>OAMdomain:adminServer' at<br/>weblogic.rmi.internal.BasicOutboundRequest.sendReceive(Basic<br/>OutboundRequest.java:109) at<br/>weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.j<br/>ava:127) at...</pre> <p>The shutdown command from <code>weblogic.Admin</code> was successful.</p> <p>The problem was solved by catching the exception that occurs when shutting down Managed Server when Administration Server is down.</p> |

---

CR079354

In WebLogic Server 6.1 SP02 a Java client accessing a DataSource via the Apache plug-in received a TunnelDeadResponse .

WebLogic Server was running under running under Windows2000, and had patch CR061847. The plug-in was running under Solaris.

The exception was:

```
<May 31, 2002 4:16:35 PM PDT> <Error>
<HTTPClientJVMConnection> <java.net.ProtocolException:
Tunneling result not OK, result: 'DEAD', id: '2' at
weblogic.rjvm.http.HTTPClientJVMConnection.receiveAndDispatch (HTTPClientJVMConnection.java:413) at
weblogic.rjvm.http.HTTPClientJVMConnection.execute (HTTPClientJVMConnection.java:296) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)>
```

When the plug-in was removed, the exception did not occur.

Analysis revealed that WebLogic Server could not find an RJVM for the host/port combination, although one existed, because the rjvm manager's synonym cache did not maintain the port information. When WebLogic Server failed to find a match in the cache when bootstrapping, it created a new RJVM.

After bootstrapping, WebLogic Server identified the new RJVM as a duplicate, and attempted to close it. In some cases, the message was not delivered to the server if queued. This caused the server to timeout on the RJVM and send a DEAD response.

This problem was resolved by a code change to make the synonym cache maintain port information.

---

CR089470

In WebLogic Server 6.1 SP03, Node Manager threw NumberFormatException when a client socket connects and then closes:

```
<Oct 28, 2002 3:38:18 PM CST> <Info>
<NodeManager@localhost:5555>
<SecureSocketListener: listening on localhost:5555>
java.lang.NumberFormatException: null
    at java.lang.Integer.parseInt(Integer.java:382)
    at java.lang.Integer.parseInt(Integer.java:463)
    at
weblogic.nodemanager.SocketInputHandler.run(SocketInputHandler.java:109)
```

There were no discernible side effects. Node Manager could still start and stop Managed Servers after the error.

A code fix was implemented to handle exceptions on sockets that are opened and closed with out any transfer of data.

CR070887 In WebLogic Server 6.1 with or without SP02, `setInstanceFollowRedirects()` does not work.

JDK1.3.1 introduced the following methods in `java.net.HttpURLConnection`: `getInstanceFollowRedirects()` and `setInstanceFollowRedirects()`.

These methods can be used to disable URL redirects for a specific `HttpURLConnection`. `HttpURLConnection` was ignoring the flag set by `setInstanceFollowRedirects()`.

The problem was solved by a code fix to `HttpURLConnection`'s redirect logic to ensure process redirects in accordance with the setting of `InstanceFollowRedirects`.

---

CR077108 During load testing of a 35-plus node cluster, a null pointer exception was encountered at `weblogic.utils.collections.NumericValueHashtable.containsKey(NumericValueHashtable.java:138)`

```
<May 9, 2002 5:08:22 PM BST> <Error> <Management>
<InvocationTargetException getting attribute
SecondaryDistributionNames on MBean bluemartini:Location
=WebConnectaw01,Name=WebConnectaw01,ServerRuntime=WebConnect
aw01,Type=ClusterRuntime. Method: public java.lang.String[]
weblogic.cluster.ClusterRuntime.getSecondaryDistributionName
s() java.lang.NullPointerException at weblogic.utils.
collections.NumericValueHashtable.containsKey(NumericValueHa
shtable.java:138) at weblogic.cluster.replication.
ReplicationManager.getSecondaryDistributionNames(Replication
Manager.java:1245) at weblogic.cluster.ClusterRuntime.
getSecondaryDistributionNames(ClusterRuntime.java:100) at
java.lang.reflect.Method.invoke(Native Method) at
weblogic.management.internal.DynamicMBeanImpl.getAttribute(D
ynamicMBeanImpl.java:511) at weblogic.management.internal.
DynamicMBeanImpl.getAttribute(DynamicMBeanImpl.java:477) at
com.sun.management.jmx.MBeanServerImpl.getAttribute(MBeanSer
verImpl.java:1181) at com.sun.management.jmx.MBeanServer
Impl.getAttribute(MBeanServerImpl.java:1151) at
weblogic.management.internal.RemoteMBeanServerImpl_WLSkel.in
voke(Unknown Source) at weblogic.rmi.internal.BasicServer
Ref.invoke(BasicServerRef.java:298) at weblogic.rmi.
internal.BasicServerRef.handleRequest(BasicServerRef.java:26
7) at weblogic.rmi.internal.BasicExecuteRequest.execute
(BasicExecuteRequest.java:22) at weblogic.kernel.Execute
Thread.execute(ExecuteThread.java:139) at weblogic.kernel.
ExecuteThread.run(ExecuteThread.java:120)
```

Analysis determined that the `getSecondaryDistributedNames()` method should allow for null value from `getOtherHost()`. Problem resolved.

---

CR079354

In WebLogic Server 6.1 SP02 and patch CR061847, a Java swing client obtaining context and accessing a JDBC data source received a TunnelDeadResponse. The client gets an initial context, gets a connection via a datasource, closes the initial context, closes the connection, and then repeats this sequence of processing. This error results:

On the server, this exception was thrown:

```
####<Nov 21, 2002 12:24:20 PM CST> <Debug>
<HTTPServerJVMConnection> <hpwlinc03> <admin> <ExecuteThread:
'14' for queue: 'default'> <> <> <000000> <Closing JVM socket:
'weblogic.rjvm.http.HTTPServerJVMConnection@4cfbd8 - id: '0',
closed: 'true', lastRecv: '1037903025816'> java.lang.
Throwable: Stack trace at weblogic.rjvm.http.HTTPServer
JVMConnection.close(HTTPServerJVMConnection.java:383) at
weblogic.rjvm.ConnectionManager.removeConnection(ConnectionM
anager.java:879) at weblogic.rjvm.ConnectionManager.
shutdown(ConnectionManager.java:508) at weblogic.rjvm.
ConnectionManagerServer.shutdown(ConnectionManagerServer.jav
a:443) at weblogic.rjvm.RJVMImpl.peerGone(RJVMImpl.java:804)
at weblogic.rjvm.RJVMImpl.getExceptionReceiving
(RJVMImpl.java:533) at weblogic.rjvm.ConnectionManager.get
ExceptionReceiving(ConnectionManager.java:736) at
weblogic.rjvm.http.HTTPServerJVMConnection.checkIsDead(HTTPSe
rverJVMConnection.java:238) at weblogic.rjvm.http.HTTPServer
JVMConnection$TunnelScavenger.trigger(HTTPServerJVMConnectio
n.java:405) at weblogic.time.common.internal.
ScheduledTrigger.executeLocally(ScheduledTrigger.java:238)
at weblogic.time.common.internal.ScheduledTrigger.
execute(ScheduledTrigger.java:229) at weblogic.kernel.Execute
Thread.execute(ExecuteThread.java:139) at weblogic.kernel.
ExecuteThread.run(ExecuteThread.java:120)
```

The client threw this exception:

```
<Nov 21, 2002 12:24:18 PM CST> <Error> <HTTPClientJVM
Connection> < java.net.Protocol Exception: Tunneling
result not OK, result: 'DEAD', id: '0' at weblogic.rjvm.
http.HTTPClientJVMConnection.receiveAndDispatch(HTTPCli
entJVMConnection.java:413) at weblogic.rjvm.http.HTTPClient
JVMConnection.execute(HTTPClientJVMConne ction.java:296) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at weblogic.kernel.ExecuteThread.run(ExecuteThread.
java:120)
```

Diagnosis revealed that a timeout occurred after WebLogic Server failed to find an RJVM for host/port combination, although it existed. The RJVM's synonym cache did not maintain port information. The problem was solved with a code fix to maintain port information in the RJVM's synonym cache.

CR080822      In the Solaris distribution with Performance Pack, log messages showed an incorrect value for the file descriptor soft limit. The value of file descriptor hard limit was shown for both the soft limit and hard limit.

This is the error in `weblogic.log`:

```
####<Mar 5, 2002 5:07:27 PM PST> <Info> <Posix Performance  
Pack> <bolinas> <myserver> <ListenThread> <system> <> <000000>  
<System has file descriptor limits of- soft: '1024', hard:  
'1024'>
```

```
####<Mar 5, 2002 5:07:27 PM PST> <Info> <Posix Performance  
Pack> <bolinas> <myserver> <ListenThread> <system> <> <000000>  
<Using effective file descriptor limit of: '1024' open  
sockets/files.>
```

This error was exhibited under Solaris 2.6 and Solaris 2.8.

The error was corrected by a syntax correction to associated script template. The incorrect syntax:

```
[ ! $? -a "$maxfiles" != 1024 ];
```

was replaced by:

```
[ ! $? -a "$maxfiles" != 1024 ];
```

---

CR085259      Session data was replicated across two separate WebLogic Clusters, hosted in a common domain, when each cluster used the same name for the session cookie. Problem was solved by implementing a check to verify that the secondary host/port is in the same cluster as the primary host for the session.

---

CR086425      Duplicate of CR085259. See above.

---

CR086758

In a multi-tier implementation, after ten hours of stress testing, the web tier hung. This occurred while the web tier (consisting of servlets, jsp's, and custom classes that implement a custom cache) was communicating with the EJB tier (all stateless session beans), after the EJB tier closed the Connection Manager, due to missed RJVM heartbeats.

Before the web tier hangs the following are the exceptions are thrown in the ejb tier.

```
<Sep 24, 2002 8:43:58 AM PDT> <Info> <RJVM> <Failure in
heartbeat trigger for RJVM:
'7831636024374910916S:10.10.10.187:[8001,8001,8002,8002,8001
,8002,-1]:webserver:sourcingWebserver'
java.rmi.ConnectException: The connection manager to
ConnectionManager for: 'weblogic.rjvm.RJVMImpl@3bedf2 - id:
'7831636024374910916S:10.10.10.187:[8001,8001,8002,8002,8001
,8002,-1]:webserver:sourcingWebserver' connect time: 'Tue Sep
24 06:15:16 PDT 2002'' has already been shut down at
weblogic.rjvm.ConnectionManager.getOutputStream(ConnectionMa
nager.java:1348) at
weblogic.rjvm.ConnectionManager.createHeartbeatMsg(Connectio
nManager.java:1306) at
weblogic.rjvm.ConnectionManager.sendHeartbeatMsg(ConnectionM
anager.java:497) at
weblogic.rjvm.RJVMImpl$HeartbeatChecker.trigger(RJVMImpl.jav
a:1032) at
weblogic.time.common.internal.ScheduledTrigger.executeLocall
y(ScheduledTrigger.java:238) at
weblogic.time.common.internal.ScheduledTrigger.execute(Sched
uledTrigger.java:229) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

Problem was solved by adding logic to assure that pending responses are notified of peer gone events.

---

CR087808

In WebLogic Server 6.1 SP03 on a Sunblade 100 single-CUP Solaris machine, two independent server instances could not look up InitialContext on each other. After one server instance looked up InitialContext, when a second server instance on the the same machine tried to look up InitialContext on the same machine this exception resulted:

```
<2002/10/09 4:23:56:JST> <Debug> <ConnectionManager> <Attempt
to sendMsg using a closed connection>
javax.naming.CommunicationException. Root exception is
java.rmi.ConnectException: Attempt to sendMsg using a closed
connection at
weblogic.rmi.internal.BasicOutboundRequest.sendReceive(Basic
OutboundRequest.java:85) at
weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwa
reRemoteRef.java:262) at
weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwa
reRemoteRef.java:229) at
weblogic.rmi.internal.ProxyStub.invoke(ProxyStub.java:35) at
$Proxy45.lookup(Unknown Source) at
weblogic.jndi.internal.WLContextImpl.lookup(WLContextImpl.ja
va:341) at
javax.naming.InitialContext.lookup(InitialContext.java:345)
at
com.bea.samples.servlet.TestServlet.service(TestServlet.java
:40) at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:265) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:200) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet
(WebAppServletContext.java:2546) at
weblogic.servlet.internal.ServletRequestImpl.execute(Servlet
RequestImpl.java:2260) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

The problem was not replicated on Sun Enterprise. The problem did not occur on Sun Blade, if the lookup was done using an IP address instead of localhost.

Analysis indicated that client RJVM tried to close duplicate T3JVMConnections. It issued a CMD\_REQUEST\_CLOSE to the server and closed the RJVM. However, if the server queued this message, then the RJVM was marked as closed.

The problem was solved by a code change to prevent RJVM shutdown when detecting duplicate connections and CMD\_REQUEST\_CLOSE does not get delivered to server.

CR087944      Running under Windows XP resulted in a `java.lang.UnsatisfiedLinkError: no muxer in java.library.path` error.

This is because WebLogic Server does not correctly report Windows XP as the host operating system. With `JDK 1.3.1_03`, `os.name` is returned "Windows 2000".

Modification of a method in the `SocketMuxer` resolved the problem..

---

CR088022      After starting Managed Servers with Node Manager, information about communications between the Administration Server and Managed Servers, such as the network channel used and the JVM ID, could not be viewed by right-clicking the server name in the left pane of the Administration console and selecting "View connections".

Analysis revealed that the `getConnections()` method of `ServerRuntimeMBean` was returning zero connection object instances, due to an underlying bug in `ConnectionManagerServer`.

The problem was resolved by a code fix to ensure that connection manager reports on connections for which the RJVM is null.

---

CR088056      WebLogic Server muxer libraries did not have build dates, making it difficult to determine which version of a library was in use. Although `NTSocketMuxer` had the build date/time embedded, `PosixSocketMuxer` did not.

Problem solved by adding build date/time to `PosixSocketMuxer`. Now, the following message will be displayed when the muxer initializes:

```
<Oct 15, 2002 3:44:04 PM PDT> <Info> <socket> <000406>
<PosixSocketMuxer was built on Oct 15 2002 15:05:11>
```

---

CR089060      Under OS/390 Linux (SuSE Linux Kernel 2.2), `libmuxer.so` threw this error:

```
java.lang.UnsatisfiedLinkError:
/opt/weblogic6/lib/linux/s390/libmuxer.so:
/opt/weblogic6/lib/linux/s390/libmuxer.so: ELF file machine
architecture not s390 at java.lang.ClassLoader$NativeLibrary.
load(Native Method) at java.lang.ClassLoader.loadLibrary0
(ClassLoader.java:1799)
```

The problem was resolved by providing binaries for the Linux Kernel 2.2.

---

CR089144

Deployment of an EJB with a custom call router in the jar failed with `IllegalArgumentException`. The class failed to load because the `classForName()` method of the `ReplicaAwareInfo` class looked for classes in the system classloader instead of the `ContextClassLoader` on the current thread.

Stacktrace:

```
java.lang.reflect.InvocationTargetException: java.lang.  
IllegalArgumentException: Class weblogic.qa.tests.cluster.  
ejb.callrouter.BaseCallRouterImpl not found at weblogic.rmi.  
cluster.ReplicaAwareInfo.classForName(ReplicaAwareInfo.java:  
172) at weblogic.rmi.cluster.ReplicaAwareInfo.getCallRouter  
(ReplicaAwareInfo.java:132) at weblogic.rmi.cluster.Basic  
ReplicaHandler.newReplicaList(BasicReplicaHandler.java:78)  
at weblogic.rmi.cluster.BasicReplicaHandler.<init>(Basic  
ReplicaHandler.java:72) at java.lang.reflect.Constructor.  
newInstance(Native Method) at weblogic.rmi.cluster.Replica  
AwareInfo.instantiate(ReplicaAwareInfo.java:186) at  
weblogic.rmi.cluster.ReplicaAwareInfo.getReplicaHandler(Repl  
icaAwareInfo.java:117) at weblogic.rmi.cluster.ReplicaAware  
RemoteRef.initialize(ReplicaAwareRemoteRef.java:79) at  
weblogic.rmi.cluster.ClusterableRemoteRef.initialize(Cluster  
ableRemoteRef.java:28) at weblogic.rmi.cluster.Clusterable  
RemoteObject.initializeRef(ClusterableRemoteObject.java:275)  
at weblogic.rmi.cluster.ClusterableRemoteObject.onBind  
(ClusterableRemoteObject.java:150) at weblogic.jndi.  
internal.BasicNamingNode.bindHere(BasicNamingNode.java:346)  
at weblogic.jndi.internal.ServerNamingNode.bindHere(Server  
NamingNode.java:105) at weblogic.jndi.internal.BasicNaming  
Node.bind(BasicNamingNode.java:281) at ...
```

Problem was resolved by setting the context classloader to the application classloader while calling `deploy()` on `ClientDrivenBeanInfoImpl` and checking in the context classloader to load the call router class.

---

CR089454

A new ability to throttle incoming request traffic by configuring the maximum length of an execute queue has been provided. The maximum length of an execute queue can be set with the new `-D` flag, `weblogic.kernel.allowQueueThrottling`.

This capability is provided to help customers throttle "slow moving resource intensive" requests on a custom queue. Queue throttling is supported for custom application queues, not for default or WebLogic Server internal queues.

When the configured queue length is exceeded calls to `Kernel.execute()` will result in client receiving a recoverable remote exception, and a 503 response.

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR090071 | <p>A high volume of assertion failed errors occurred in <code>PosixSocketMuxer.cleanup()</code>, accompanied by a steady growth in the number of sockets in the <code>CLOSE_WAIT</code> state. <code>ulimit</code> was reached, and server instance restart was required.</p> <p>Problem was solved by implementing logic that checks if <code>fdr.sock</code> is null before throwing an assertion error in <code>cleanup</code>.</p>                                                                                                                                                                                                  |
| CR090341 | <p>Erroneous failure duration was reported in <code>t3.srvr.ListenThread</code> calling <code>logListenFailed()</code>. This was a numeric error in a message generated when the server failed to listen due to an underlying <code>IOException</code>. Example of the message:</p> <pre>&lt;Jun 11, 2002 2:37:33 PM PDT&gt; &lt;Critical&gt; &lt;WebLogicServer&gt;<br/>&lt;Failed to listen on port 7772, failure count: 3, failing for<br/>1,023,831,450 seconds, java.net.SocketException: File table<br/>overflow&gt;</pre> <p>The problem was fixed by correcting a typographical error in <code>t3.srvr.ListenThread</code>.</p> |
| CR090823 | <p>JSPs were recompiled unnecessarily for the e2e sample. The JSPs in e2edomain always got recompiled when running b2c and b2b examples. The time stamps of those JSPs were properly backdated, however.</p> <p>This problem resulted from a bug in the <code>JspStub.getClassLoader</code> method. Problem resolved by fixing the bug.</p>                                                                                                                                                                                                                                                                                             |
| CR091420 | <p>WebLogic Server 6.1 SP04 failed to start if the "Enable Post-Bind UID" option for a Unix Machine was checked. (This attribute can be configured in the Machines node of the Administration Console; its purpose is to return the Unix UID a server instance running a UNIX machine will run under after it has carried out all privileged startup actions).</p> <p>The problem was corrected with a code fix.</p>                                                                                                                                                                                                                    |

CR092704 In WebLogic Server 6.1 SP02, hangs occurred frequently because socket reader threads were blocked. The thread that owned the POLL lock was attempting to close an SSL socket, but could not progress because it could not obtain the lock on the output stream required in the `sendRecord()` method.

Stack trace:

```
"ExecuteThread: '23' for queue: 'default'" daemon prio=5
tid=0x6d6c40 nid=0x24 waiting for monitor entry
[0xe4e81000..0xe4e81a28] at
weblogic.security.SSL.SSLSocket.sendRecord(SSLSocket.java:10
49) at
weblogic.security.SSL.SSLSocket.sendAlert(SSLSocket.java:100
7) at
weblogic.security.SSL.SSLSocket.close(SSLSocket.java:1153)
at
weblogic.security.SSL.SSLSocket.close(SSLSocket.java:1141)
at
weblogic.socket.SocketMuxer.closeSocket(SocketMuxer.java:236
```

A code fix was implemented to ensure that SSL sockets do not write data to sockets on close for abortive shutdowns.

---

CR092933 Thread dump error occurred with this configuration:

- VM: java.version='1.3.1\_02'
- os.name='windows 2000'
- java.vendor.url='http://java.sun.com/'

Thread dumps failed because the `JVM_DumpAllStacks` symbol had not been globally declared for Hotspot client and Version 1.3.1\_x virtual machines.

Thread dumps using `weblogic.Admin` are now possible with 1.3.1\_X hotspot client/server JVMs.

---

CR093416

An `EmptyStackException` was thrown by a Managed Server when the Controller servlet was being deployed.

The following error is thrown by a managed server when the Controller servlet is being deployed:

```
[WebAppServletContext(35527,btn,/btn)] Servlet failed with
Exception> java.util.EmptyStackException at weblogic.utils.
collections.Stack.pop()Ljava.lang.Object;(Unknown Source) at
weblogic.kernel.ResettableThreadLocalStack.pop()Ljava.lang.O
bject;(Unknown Source) at weblogic.jndi.internal.
ThreadEnvironment.pop()Lweblogic.jndi.Environment;(Unknown
Source) at weblogic.jndi.internal.WLContextImpl.close()V
(Unknown Source) at weblogic.jndi.factories.java.ReadOnly
ContextWrapper.close()V(Unknown Source) at be.btn.util.
JndiUtil.getEnv(Ljava.lang.String;)Ljava.lang.Object;(Unknow
n Source) at be.btn.web.HttpControllerServlet.init()V(Unknown
Source) at javax.servlet.GenericServlet.init(Ljavax.servlet.
ServletConfig;)V(Unknown Source) at weblogic.servlet.
internal.ServletStubImpl.createServlet()Ljava.servlet.Serv
let;(Unknown Source) at weblogic.servlet.internal.
ServletStubImpl.createInstances()V(Unknown Source) at...
```

The problem was related to the `btn.utils.JndiUtils.getEnv()` method—it closed the context that it got on lookup—`java:com/env`—so `javaURLContextFactory` was not pushing the environment.

Problem resolved.

---

CR094101

There was an interoperability issue between WebLogic Server 6.1 SP02 and WebLogic Server 7.0 SP01. An MDB in WebLogic Server 6.1 called an EJB in WebLogic Server 7.0. The EJB is timed out and threw a `weblogic.transaction.TimedOutException`, which is new in WebLogic Server 7.0 and not available in WebLogic Server 6.1. The WebLogic Server 6.1 MDB `onMessage()` method caught the exception, and threw a `ClassNotFoundException` when the code `Exception.printStackTrace()` was executed.

The problem was solved by adding a `weblogic.transaction.TimedOutException` class to prevent `ClassNotFoundException` when interoperating with WebLogic Server 7.0 servers.

---

## 6 Resolved Problems

---

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR094724 | <p>In WebLogic Server 6.1 SP03 and later, WebLogic Server native libraries on HP-UX were not being compiled using <code>ccfront</code>, instead of <code>aCC</code>. <code>aCC</code> is the ANSI C compiler which replaced <code>ccfront</code> in 1998.</p> <p>As a result incompatible runtime libraries were loaded (<code>libc.2</code> from <code>ccfront</code> compilation, and <code>libcSup.2</code> from SUN's Java, compiled with <code>aCC</code>). Incompatible runtime libraries can cause crashes.</p> <p>The problem was solved by changing WebLogic Server builds to use <code>aCC</code> for <code>hpux11</code></p> |
| CR095267 | Duplicate of " <a href="#">CR094561</a> ".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| CR095487 | Duplicate of " <a href="#">CR087808</a> ".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| CR095949 | <p>In WebLogic Server 6.1 SP04, when <code>FailureIsFatal</code> option for a startup class is true and the startup class fails, WebLogic Server was not shutdown as expected.</p> <p>The <code>StartupClassRunner</code> is invoked before and after applications are deployed. Analysis revealed that if a startup class failed before application deployment, and was marked <code>failureIsFatal</code>, the <code>fatalException</code> was lost.</p> <p>The problem was solved by a code fix to ensure that the <code>fatalException</code> is not lost, and the the server instance is shutdown appropriately.</p>               |
| CR096114 | <p>The change associated with "<a href="#">CR092933</a>" which enables thread dump redirection in 1.3.1_x versions of Sun JDK, did not work in NT Services because the <code>stdio</code> handles were invalid.</p> <p>The problem was resolved by a code change, to create <code>stdio</code> handles in <code>beasvc</code>.</p>                                                                                                                                                                                                                                                                                                      |
| CR101322 | <p>A customer used a custom realm that makes an outbound RMI call. This happens from <code>BootServicesImpl.invoke()</code> from within the RJVM layer on a reader thread. when many such calls are made, reader threads were blocked making outbound calls leaving no threads for reading the response.</p> <p>The problem was solved by moving <code>BootServices</code> into the RMI layer so that it can dispatch to the default execute queue.</p>                                                                                                                                                                                 |
| CR102021 | <p>In WebLogic Server 6.1 SP04, under load test, <code>PosixMuxer</code> threw the following exceptions</p> <pre>&lt;Nov 21, 2002 9:38:29 AM EST&gt; &lt;Error&gt; &lt;socket&gt; &lt;000421&gt; &lt;Uncaught Throwable in processSockets java.io.IOException: unexpected exception in poll: (4) Interrupted system call java.io.IOException: unexpected exception in poll: (4) Interrupted system call at weblogic.socket.PosixSocketMuxer.poll(Native Method) at...</pre> <p>The problem was solved with a code change to restart <code>poll</code> if interrupted.</p>                                                               |

---

CR102058

Use of URLClassLoader on the client on a remote method call resulted in a NoClassDefFoundError.

```
c:\java\java131_07\bin\java -classpath
".;client.jar;C:\home\ravia\weblogic\dev\src700\3rdparty\web
logicaux.jar" URLTest qa146 9901 ejb20-statefulSession-
TraderHome installadministrator installadministrator
java.lang.NoClassDefFoundError:
weblogic/rmi/extensions/server/Stub at
java.lang.ClassLoader.defineClass0(Native Method) at
java.lang.ClassLoader.defineClass(ClassLoader.java:488)
at...
```

Analysis revealed that when creating AugmentableSystemClassLoader, it was always set to SystemClassLoader. This caused the client to use its system classloader, resulting in the NoClassDefFoundError.

The problem was solved by setting ThreadContextClassLoader as parent for AugmentableSystemClassLoader.

---

## Deployment

---

| <b>Change Request Number</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR089031                     | <p>After an upgrade from 6.1 SP02 with cr058358_61sp2.jar to SP03, a degradation in deployment performance occurred on a Managed Server. Degradation was not exhibited on the Administration Server.</p> <p>The problem was related to the <code>weblogic.j2ee.Component.getModuleMBean()</code> method, which was implemented for CR058358. Because <code>ApplicationDescriptorMBean</code> is only registered in the Administration Server, any requests to that mbean from a Managed Server resulted in RMI calls to the Administration Server.</p> <p>Problem was solved by building a local cache mbean (<code>ApplicationDescriptorMBeanImpl_Cached</code>) on the Managed Server. Now, the first method request goes to the Administration Server to retrieve the information and save locally. Subsequent requests are served locally, and the cache is invalidated when the application is undeployed.</p>                                                                                                                                                                                                                |
| CR089765                     | <p>Executing <code>weblogic.refresh</code> on Windows 2000 to update files (JSP or HTML) to WebLogic Server Administration Server and Managed Server running on Linux caused a <code>java.util.zip.ZipException</code>.</p> <p>The ZIP exception happens when WebLogic Server is configured as an Administration Server or Managed Server and the Web Application for which the <code>weblogic.refresh</code> is done is targeted to the Managed Server.</p> <p>When the Managed Server tries to get the refreshed files from the Administration Server, exceptions are thrown by both server instances.</p> <p>Exhibited in this configuration:</p> <ul style="list-style-type: none"><li>■ WebLogic Server 61SP03 on Linux (Redhat, JDK1.3.1) (1 admin, 1 managed)</li><li>■ Win2K, calling <code>weblogic.refresh</code> (WebLogic Server61SP3, JDK1.3.1)</li></ul> <p>Problem diagnosis revealed an error in the file location path for the refreshed file from Managed Server to Administration Server, The problem was corrected by a correction to <code>FileDistributionServlet.doGetMultipleJspRefreshRequest</code>.</p> |

---

CR091897

In WebLogic Server 6.1 SP04, deployment failed for a web application with a short name, such as like `ab.war` or `i.war`. This error occurred:

```
java.lang.IllegalArgumentException: Prefix string too short
at java.io.File.createTempFile(File.java:1232) at
weblogic.j2ee.Component.retrieveComponent(Component.java:296)
at weblogic.j2ee.Component.<init>(Component.java:188) at
weblogic.j2ee.WebAppComponent.<init>(WebAppComponent.java:45)
at weblogic.j2ee.Application.addComponent(Application.
java:149) at weblogic.j2ee.J2EEService.addDeployment
(J2EEService.java:117)...
```

The bug was corrected.

---

CR097560

In WebLogic Server 6.1 SP04, when custom `InitialContextFactory` was set to `weblogic.jndi.Environment`, `getInitialContext()` instead used `DEFAULT_INITIAL_CONTEXT_FACTORY`.

The problem was corrected by a code fix to ensure that the `weblogic.jndi.Environment` object considers `InitialContextFactory`, if specified.

---

## EJB

---

| <b>Change Request Number</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR063275                     | <p>WebLogic Server EJB did not provide the ability to write a finder method on a <code>byte []</code> field. This was non-compliant with the J2EE EJB-QL definition: “The allowable types in EJB QL are the abstract schema types of entity beans and dependent objects, the defined types of <code>cmp</code>-fields, and the entity object types of remote entity beans.” This problem was identified in WebLogic Server 6.1 SP01. This error occurred at build time:</p> <pre>ejbc:<br/>[java]<br/>[java] ERROR: Error from ejbc: Error while reading<br/>'META-INF/weblogic-cmp-rdbms-jar.xml'. The error was:<br/>[java]<br/>[java] invalid query: In EJB containerManaged, for a query<br/>defined in the ejb-jar.xml file with a method signature,<br/>findFk(byte[]), we failed to find a corresponding method in<br/>the remote home interface, local home interface, or bean class<br/>that matches this signature. Note that class parameters such<br/>as java.lang.String must be fully qualified, thus 'String'<br/>would not match 'java.lang.String'. [java]</pre> <p>This problem has been fixed. WebLogic Server EJB now supports finder methods on a <code>byte []</code> field.</p> |
| CR063837                     | <p>In WebLogic Server 6.1 SP02, the following error was thrown when the EJB Container tried to verify the existence of database tables and columns that did not exist:</p> <pre>weblogic.utils.AssertionError: ***** ASSERTION FAILED *****<br/>Table: Cmp_Birthday Full Table Check failed, but table all<br/>columns were found! ] at<br/>weblogic.ejb20.utils.TableVerifier.verifyTableAndColumnsExist<br/>(TableVerifier.java:371) at<br/>weblogic.ejb20.utils.TableVerifier.verifyTableExistsAndCreat<br/>eMaybe(TableVerifier.java:391)</pre> <p>The problem was solved with a code fix.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

---

CR076272

When hot deploying an EJB, compilation errors were not helpful. For instance:

```
<30 avr. 02 11:03:14 BST> <Error> <Management> <Error
deploying
application.\config\mydomain\applications\ldaprofile.jar:ja
va.lang.reflect.UndeclaredThrowableException>
```

Error reporting is now improved for hot-deployed EJBs. Now, run-time exceptions and errors are reported similarly to the method used in standalone ejbc. The server log and stdout will provide information to help the user understand the failure. For instance:

```
<Oct 1, 2002 4:37:18 PM PDT> <Error> <J2EE> <Error deploying
application ldaprofile: Unable to deploy EJB: ldaprofile.jar
from ldaprofile.jar: java.lang.NoClassDefFoundError:
com/bea/p13n/property/EntityPropertyManager at
java.lang.ClassLoader.defineClass0(Native Method) at
java.lang.ClassLoader.defineClass(ClassLoader.java:486) at
java.security.SecureClassLoader.defineClass(SecureClassLoade
r.java:111) at weblogic.utils.classloaders.GenericClass
Loader.findLocalClass(GenericClassLoader.java:394) at
weblogic.utils.classloaders.GenericClassLoader.findClass(Gen
ericClassLoader.java:157) at
java.lang.ClassLoader.loadClass(ClassLoader.java:297) at
java.lang.ClassLoader.loadClass(ClassLoader.java:253)
at.....
```

CR076386

In 6.1 SP03, using a cmp-ejb with cmr-fields only generated incorrect SQL insert statements. This problem was exhibited with MS-SQL Server 2000 with Automatic-Primary-Key-Generation-TPYE : SQL\_SERVER. The Container Generated SQL-Statement:

```
Insert into tblLicence(, 5 , 7) values (,licCompID,licOrgID)
where compID and orgID are foreign keys in the table which corresponds to this entity
bean. After adding an cmp-field, the unmodified (empty) ejbCreate-Method
generates the correct SQL:
```

```
Insert into tblLicence(licArchivNR,licCompID,licOrgID)values
(null , 5 , 7)
```

The problem has been resolved, the container no longer generates an incorrect SQL Insert statement when the bean has only (pk) field and cmr fields.

---

## 6 Resolved Problems

---

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR080331 | <p>In 6.1 SP03, deploying a version 5.1 EJB jar using the Administration Console resulted in a <code>ProcessorFactoryException</code>:</p> <pre>Unable to deploy EJB: TxRolledbackExceptionBean.jar from TxRolledbackExceptionBean.jar:  The XML parser encountered an error in your deployment descriptor. Please ensure that your DOCTYPE is correct. You may wish to compare your deployment descriptors with the WebLogic Server examples to ensure the format is correct. The error was:  weblogic.xml.process.ProcessorFactoryException: The public id, "-//BEA Systems, Inc.//DTD WebLogic 5.1.0 EJB//EN ", specified in the XML document is invalid. Use one of the following valid public ids:  "-//BEA Systems, Inc.//DTD WebLogic 5.1.0 EJB//EN" "-//BEA Systems, Inc.//DTD WebLogic 6.0.0 EJB//EN"</pre> <p>This problem occurred because WebLogic Server 5.1 supported white space in the DTD declaration for deployment descriptors.</p> <p>The problem was solved by a code change to trim the extra spaces in the key string before searching for the parser in the hash table with this key.</p> |
| CR083239 | <p>The EJB QL NOT MEMBER clause no longer causes infinite looping.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| CR083240 | <p>The EJB QL NOT MEMBER clause in conjunction with WHERE no longer creates a bad query.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| CR084978 | <p>Added the Extra EJBC Options field to the Administration Console, so that heap size and other options can now be passed to ejbc via the Console.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| CR087151 | <p>For a stateless session bean, when the <code>weblogic-ejb-jar.xml</code> contains:</p> <pre>&lt;stateless-bean-is-clusterable&gt;False&lt;/stateless-bean-is-clusterable&gt;</pre> <p>and the bean was deployed on a cluster, this error was generated:</p> <pre>&lt;Mar 14, 2003 3:35:00 PM PST&gt; &lt;Error&gt; &lt;Cluster&gt; &lt;Conflict start: You tried to bind an object under the name ejb20-statelessSession-TraderHome_EO in the JNDI tree. The object you have bound from 172.17.24.112 is non clusterable and you have tried to bind more than once from two or more servers. Such objects can only be deployed from one server.&gt;</pre> <p>This problem was resolved by a code fix to ensure that JNDI bindings for non-clustered stubs are not replicated.</p>                                                                                                                                                                                                                                                                                                                              |

---

---

CR088526

In WebLogic Server 6.1 SP03, the EJB container breaks the EJB2.0 contract: "Tx not set to rollback when using BMT and exception is thrown".

Problem was exhibited with an MDB with BMT where the `onMessage` method throws an exception. The container did not handle the exception in accordance with the EJB2.0 specifications (Section 18.3.2 Table 18.) The container should log the exception (done) delete the bean instance (done), and mark the transaction for rollback (NOT done). As a result a transaction remained associated with the thread.

A code fix was implemented to resolve this problem.

---

CR089759

WebLogic Server 6.1:SP03: [EJB] : When calling

context.getCallerPrincipal() from ejbStore() using UserTransaction context, the following exception is thrown:

```
<Oct 24, 2002 5:52:24 AM PDT> <Info> <EJB> <Exception from
ejbStore:javax.ejb.EJBException: ejbStore: nulljavax.ejb.
EJBException: ejbStore: null at AccountBean.ejbStore(Account
Bean.java:99)atAccountBean_t4qgrab_Impl.ejbStore
(AccountBean_t4qgrab_Impl.java:131)at weblogic.ejb20.manager.
DBManager.storeBean(DBManager.java:266) at weblogic.ejb20.
manager.DBManager.beforeCompletion(DBManager.java:397) at
weblogic.ejb20.internal.TxManager$TxListener.beforeCompleti
on(TxManager.java:494) at weblogic.transaction.internal.
ServerSCInfo.callBeforeCompletions(ServerSCInfo.java:551) at
weblogic.transaction.internal.ServerSCInfo.startPrePrepareAn
dChain(ServerSCInfo.java:88)at weblogic.transaction.
internal.ServerTransactionImpl.localPrePrepareAndChain(Serve
rTransactionImpl.java:979)at weblogic.transaction.
internal.ServerTransactionImpl.globalPrePrepare(ServerTransa
ctionImpl.java:1503) at weblogic.transaction.internal.
ServerTransactionImpl.internalCommit(ServerTransactionImpl.j
ava:215) at weblogic.transaction.internal.Server
TransactionImpl.commit(ServerTransactionImpl.java:189)at
weblogic.transaction.internal.CoordinatorImpl.commit(Coordin
atorImpl.java:68) at weblogic.transaction.internal.
CoordinatorImpl_WLSkel.invoke(Unknown Source)at weblogic.rmi.
internal.BasicServerRef.invoke(BasicServerRef.java:305)at
weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServ
erRef.java:274) at weblogic.rmi.internal.BasicExecute
Request.execute(BasicExecuteRequest.java:22)
atweblogic.kernel.ExecuteThread.execute(ExecuteThread.java:1
39) at weblogic.kernel.ExecuteThread.run(Execute
Thread.java:120)><Oct 24, 2002 5:54:53 AM PDT> <Info>
<Management> <Configuration changes for domain saved to the
repository.>
```

This problem was a result of changes introduced in SP03 to comply with section 21.2.5.1 of the EJB specification. The problem was corrected.

CR089953

In SP03, BEA tests of `ejb20.locks.ExclusiveLockManager` indicated a memory leak. The memory leak appears as a result of deploying and undeploying. Analysis revealed that the timer/trigger in `DiskSwap` was not cancelled when an EJB was undeployed, leaving a reference in `TimeGenerator`, and that `ExclusiveLockManager` buckets were not being garbage collected.

Problem was solved by code fix to cancel the time/trigger appropriately and ensure proper garbage collection.

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR090143 | <p>The default pool-size for MDBs was larger than the thread pool size default, resulting in MDB instances consuming the entire default thread pool and blocking waiting for non-MDB work to complete—leaving no threads left to do the unblocking work.</p> <p>Problem was resolved by, when an MDB is assigned to the main thread pool, limiting MDB pool size to a percentage of ((thread pool size) - (socket reader threads)).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| CR090515 | <p>The Administration Console reported incorrect value for the number of waiters for Entity beans. After clients had timed out, the console reported that there were waiters.</p> <p>The problem was resolved by by a correction to <code>ExclusiveLockManager</code> to decrement the <code>waiterTotalCount</code> when it is done waiting.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR091436 | <p>In SP03, <code>JMSConnectionPoller</code> did not close its initial contexts. When an MDB used external <code>InitialContextFactory</code> for lookups and the JMS provider was not accessible, the connections and other resources accumulated rapidly.</p> <p>The problem was corrected with a code fix to close the context when it is done for active memory clean up.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR091722 | <p>In SP03, under heavy loads, an MDB threw an <code>IllegalStateException</code> when resuming a transaction after invoking a business method; the JVM crashes when the transaction is being resumed.</p> <p>This behavior was exhibited with these application characteristics: a client sends messages to a Queue and an MDB listens on it. The <code>tx.attribute</code> for this MDB is "Required". The create method of the MDB is doing a <code>home.create</code> for creating some BMT SLSBs and the <code>onMessage</code> method is invoking the business method for this SLSB. The business method of this SLSB is sends messages to two different queues.</p> <p>The following error was observed:</p> <pre>&lt;EJB Exception during invocation from home: com.gfs.corp.batch.cost.costupdate.ejb.NextOrderCostUpdatePr ocessorEJB_xtvwzj_LocalHomeImpl@54b120 threw exception: java.lang.StackOverflowError&gt; java.lang.StackOverflowError &lt;&lt;no stack trace available&gt;&gt;</pre> <p>The local interface was not handing exceptions properly.</p> <p>The problem was solved by a code fix to correct the error handling so that the container properly rolls back the transaction when exception occurs.</p> |
| CR093850 | <p>In SP04, runtime monitoring of the entity beans in reported an incorrect value for the use the cached beans current count—the value exceeded the max beans in cache specified in the deployment descriptor and also the cached beans current count.</p> <p>The value reported was incorrect, and new beans were being created inappropriately.</p> <p>A code correction was made to release beans to the pool after removal from cache.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

CR095173

The `idle-timeout-seconds` element in `weblogic-ejb-jar.xml` determines how long the EJB container waits before passivating stateful session beans, that is, removing them from cache and writing them to disk. The EJB container also used to use this element to determine how long to wait before removing passivated EJBs from the disk. Some users wanted stateful session beans to remain on disk longer than `idle-timeout-seconds`. In other words, they want to specify how long stateful session beans stay idle in the cache and how long they stay idle on disk using two different elements.

A new element has been added, `session-timeout-seconds`, which specifies how long the EJB container waits before removing an idle stateful session bean from disk.

---

CR094524

In WebLogic Server 6.1 SP03 and SP04, read-only EJBs with a many:one container-managed relationship CMR caused a `LockTimedOutException`.

```

avant getCustomerVO.getCountryVO() [ExclusiveLockManager$Lock
Bucket] : ** LOCK ACQUIRE --> WAITING -- ejb-name: Country
primary key: 002 lockClient: Name=[EJB charu.ejbrelations
final.CountryBean.getCountryVO()],Xid=11:1d0d46b2(3053175),S
tatus=Active,numRepliesOwedMe=0,numRepliesOwedOthers=0,secon
ds since begin=0,seconds left=30,activeThread=Thread
[ExecuteThread: '2' for queue: 'default',5,Thread Group for
Queue: 'default'],SCInfo[mydomain+myserver]=(state=
active),properties=( {weblogic.transaction.name=[EJB
charu.ejbrelationsfinal.CountryBean.getCountryVO()],
LOCAL_ENTITY_TX=true}),OwnerTransactionManager=ServerTM[Serv
erCoordinatorDescriptor=(CoordinatorURL=myserver+172.23.135.
56:7011+mydomain+, Resources={})]) wait (MS): 30000
[ExclusiveLockManager$LockBucket] : ** LOCK TIME OUT AFTER
WAITING -- ejb-name: Country primary key: 002 lockClient:
Name=[EJB charu.ejbrelationsfinal.CountryBean.get
CountryVO()],Xid=11:1d0d46b2(3053175),Status=Marked
rollback.[Reason=weblogic.transaction.internal.TimedOut
Exception: Transaction timed out after 29 seconds Name=[EJB
charu.ejbrelationsfinal.CountryBean.getCountryVO()],Xid=11:1
d0d46b2(3053175),Status=Active,numRepliesOwedMe=0,numReplies
OwedOthers=0,seconds since begin=29,seconds left=30,active
Thread=Thread[ExecuteThread: '2' for queue:'default',5,Thread
Group for Queue: 'default'],SCInfo[mydomain+myserver]=
(state=active),properties=( {weblogic.transaction.name=[EJB
charu.ejbrelationsfinal.CountryBean.getCountryVO()],
LOCAL_ENTITY_TX=true}),OwnerTransactionManager=ServerTM[Serv
erCoordinatorDescriptor=(CoordinatorURL=myserver+172.23.135.
56:7011+mydomain+, Resources={})])],numRepliesOwedMe=0,
numRepliesOwedOthers=0,seconds since begin=30,seconds
left=10,activeThread=Thread[ExecuteThread: '2' for queue:
'default',5,Thread Group for Queue:'default'],SCInfo
[mydomain+ myserver]=(state=active),properties=( {weblogic.
transaction.name=[EJB charu.ejbrelationsfinal.CountryBean.
getCountryVO()], LOCAL_ENTITY_TX=true}),OwnerTransaction
Manager=ServerTM[ServerCoordinatorDescriptor=(CoordinatorURL
=myserver+172.23.135.56:7011+mydomain+, Resources={})]) wait
(MS): 30000

```

The problem was resolved by modifying the `if` condition check from `rbd.isReadOnly()` to `READONLY_EXCLUSIVE_CONCURRENCY` check.

CR095545

When attempting to deploy two message-driven beans with the same `ejb-name` but different JNDI mappings on the same WebLogic Server instance, the first bean was deployed successfully, but the deployment of the second bean failed with this exception: `weblogic.management.ManagementException: - with nested exception:`

```
[javax.management.InstanceAlreadyExistsException:  
jpdomain:Location=jpserver,Name=MessageQueueHandlerBean,ServerRuntime=jpserver,Type=EJBMessageDrivenRuntime] at  
weblogic.management.runtime.RuntimeMBeanDelegate.register(RuntimeMBeanDelegate.java:96) at  
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeMBeanDelegate.java:83) at  
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeMBeanDelegate.java:53) at  
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeMBeanDelegate.java:63) at  
weblogic.ejb20.deployer.MessageDrivenRuntimeMBean.<init>(MessageDrivenRuntimeMBean.java:18) at  
weblogic.ejb20.deployer.MessageDrivenBeanInfoImpl.deploy(MessageDrivenBeanInfoImpl.java:450) at  
weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.java:1299) at  
weblogic.ejb20.deployer.Deployer.deploy(Deployer.java:1005)
```

The problem was due to non-unique `MessageDrivenRuntimeMBean` names. The problem was solved by code change to ensure creation of unique names. .

---

CR096848

In 6.1 SP04, `<is-modified-method-name>` was not called on CMP 2.0 beans. `ejbStore` was called at every bean method invocation and WebLogic Server determined afterwards if the store is to avoid. This caused performance issues in applications that frequently used `<is-modified-method-name>`.

The problem was solved by implementing the `<is-modified-method-name>` function for CMP EJB 2.0.

---

---

CR099420      The `ejbc` compiler failed with the following error when a two-dimensional array was specified as an argument:

```
Unable to set the method permission for method
"isAuthorized(java.lang.String,[[java.lang.String)]. No
matching method could be found. Please verify the method
signature specified in the ejb-jar.xml file matches that of
your EJB. ERROR:ejbc found errors.
```

The problem was solved by modifying the `makeMethodParameter()` method of the `eblogic.ejb20.deployer.mbimpl.MethodDescriptorImpl` class to generate the appropriate method parameter signature depending on the dimension of the array passed.

---

CR101384      When a `UserTransaction.commit` failed in `MDListener`, subsequent invocations of the MDB from JMS failed in `UserTransaction.begin` because there was already a transaction on the thread. This series of events occurred:

1. an MDB was consuming from a JMS queue on another server
2. JMS server died
3. Inside `weblogic.ejb20.internal.MDListener`, `UserTransaction.commit` threw an exception
4. `MDListener` goes on with its life
5. Subsequent invocations of the MDB from JMS failed in `UserTransaction.begin`

This problem was solved with a code change to make `MDListener` suspend the current transaction before starting a new one in the event that an unexpected exception prevented the prior transaction from completing.

---

CR101315

It was reported that `jsp:setProperty` parameter conversions failed.

When the `jsp:setProperty` action sets the values of properties in a bean, the parameters are converted per JSP.2.13.2.1 using the target property to determine the target type. The jsp file with the `jsp:setProperty` action fails to compile with the error messages below:

```
... setCount(int) in
weblogic.qa.xmlbasedtests.webapp_jsp_tests.FooBean cannot be
applied to (java.lang.String) ... setBool(boolean) in
weblogic.qa.xmlbasedtests.webapp_jsp_tests.FooBean cannot be
applied to (java.lang.String) ... setmyDouble(double) in
weblogic.qa.xmlbasedtests.webapp_jsp_tests.FooBean cannot be
applied to (java.lang.String) ... setFloat(float) in
weblogic.qa.xmlbasedtests.webapp_jsp_tests.FooBean cannot be
applied to (java.lang.String) ... setmyLong(long) in
weblogic.qa.xmlbasedtests.webapp_jsp_tests.FooBean cannot be
applied to (java.lang.String)
```

This behavior was the result of the changes associated with “CR098402”. (Fixed a problem with `jsp:setProperty`, request-time expression in the property value was converted to `String`. When assigning from a value given as a request-time attribute, no type conversions are applied.)

Investigation revealed that the specification says:

“When assigning from a value given as a request-time attribute, no type conversions are applied, as indicated in Section JSP.2.13.2.3.”

The code was determined to be functioning in accordance with the specification, and no changes were required to close this CR.

---

## Installer

---

| Change Request Number | Description                                                                                                                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092156              | <p>The WebLogic Server 6.1 SP04 upgrade installer build did not include latest <code>beasvc.exe</code>. For related CRs, see See “CR091420” on page 6-155, and “091420” on page 5-16.</p> <p>The SP05 upgrade installer does not exhibit this problem.</p> |

---

## JDBC

---

| Change Request Number | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR085559              | <p>When a user transaction failed to rollback because Oracle database was down, the JDBC connections used in the transaction were not released, resulting in connection pool depletion.</p> <p>This problem was exhibited under WebLogic Server 6.1 SP02, Oracle8.1.6, 8.1.7 jDriver for Oracle, and Oracle thin driver.</p> <p>Analysis revealed that the <code>internalrollback</code> method in the JTS connection was not calling the <code>internalclose()</code> method, so if the rollback failed the connection leaked. A code fix solved the problem.</p> |
| CR089713              | <p>In WebLogic Server 6.1 SP03, <code>weblogic.Admin RESET_POOL</code> did not re-initialize connections to not-reserved.</p> <p>After using <code>weblogic.Admin RESET_POOL</code>, the log indicated that all connections were refreshed, but if they were reserved before the reset, they are still reserved after the reset.</p> <p>The problem was solved with a code change to put reserved resource back on <code>resourcesFree</code> list and clear the associated flag.</p>                                                                              |

---

CR090379 In previous releases of WebLogic Server, if application code created JDBC objects that were abandoned without being closed, the objects would be lost but would still hold memory or open cursors, even after being garbage collected. If too many such objects were created, the server would eventually run out of memory and the database may run out of cursors.

In WebLogic Server 6.1SP5, the code was changed so that abandoned JDBC objects are closed before being garbage collected.

**Note:** If you immediately create a JDBC object that is identical to an abandoned object, WebLogic Server creates a clone of the original JDBC object. However, when the original leaked object is closed, the clone will also be affected.

You can avoid abandoning JDBC objects by following the best practices described in “[Closing JDBC Objects](#)” in *Programming WebLogic JDBC*.

---

CR090761 A `StringIndexOutOfBoundsException` occurred in the OCI driver. The problem occurred occasionally in multiple queries when running a CMP entity bean, always when dealing with an integral at or near (depending on the precision of the number in the database) `weblogic.jdbc.pool.ResultSet.getLong(ResultSet.java:107)`

The full exception is: `java.sql.SQLException: java.lang.StringIndexOutOfBoundsException - String index out of range: 0`

Analysis revealed that in `weblogic.db.oci.OciCurser.getValue(OciColumn,int,int,boolean,boolean)`, the methods `new Long(string)` and `new BigInteger(string)` were being called without appropriate checks, resulting in the `StringIndexOutOfBoundsException`.

A code fix solved the problem.

---

CR091577 In WebLogic Server 6.1 SP03, connection pool reset methods were unnecessarily synchronized. The `ResourceAllocator` method `resetThisOne()` was synchronized. As a result, when all execute threads performed a `testConnOnReserve`, and replaced a dead connection, they queued up, instead of establishing connections in parallel.

A code fix solved the problem.

---

CR092453      In WebLogic Server 6.1 SP04, with Oracle Thin XA with the CLASSES12.zip from Oracle 9.2, a stateless session bean calling EJB caused XAER\_PROTO after "Configuration Changes saved to the repository" message appeared.

```
START SLEEP 2: After updating thevalue to 1...
DONE SLEEP 2: After updating thevalue to 1...
START SLEEP 2: After updating thevalue to 2...
DONE SLEEP 2: After updating thevalue to 2...
START SLEEP 2: After updating thevalue to 3...
DONE SLEEP 2: After updating thevalue to 3...
START SLEEP 2: After updating thevalue to 4...
DONE SLEEP 2: After updating thevalue to 4...
START SLEEP 2: After updating thevalue to 5...
DONE SLEEP 2: After updating thevalue to 5... Current value is
5 <
Dec 6, 2002 10:26:59 PM MST> <Info> <Management>
<Configuration changes for domain saved to the repository.>
SQLException -- XA error: XAER_PROTO : Routine was invoked in
an improper context start() failed on resource 'OracleXA' null
Current value is 0
```

The problem caused by a known problem in Oracle client 9.2.0.[01], that is solved in 9.2.0.2. A code change to in WebLogic Server was implemented to work-around the 9.2.0.[01] issue.

---

CR092791      In WebLogic Server 6.1 (any SP), it was not possible to use specific Oracle objects (Array, Struct, ...) through a connection pool based on the Oracle Thin Driver. The objects returned by the Oracle thin driver were not serializable nor remote and therefore could not be passed over RMI.

A new package has been introduced : `weblogic.jdbc.vendor.oracle` that contains "proxies" for these objects and allow them to be passed through the connection pool.

---

CR093245      After calling `JDBCConnectionPoolRuntimeMBean.resetStatementProfile`, it was not possible to display SQL traces. Calling `resetStatementProfile()` cleared the trace file, but it was not possible to do traces, even after tracing was enabled with the `JdbcSqlTraceAdmin` tool. It was necessary to restart the server instance to re-enable tracing.

A code fix solved the problem.

---

- 
- CR093563      A deadlock occurred between "Finalizer" and `weblogic/jdbc/jta/Statement.java`.
- The problem was exhibited under Solaris 5.8 - WebLogic Server 6.1SP3 - java v1.3.1\_03 (build 1.3.1\_03-b03), Java HotSpot(TM) Client VM (build 1.3.1\_03-b03, mixed mode). This error occurred:
- ```
"ExecuteThread: '5' for queue: 'default': waiting to lock
monitor 0xelba8 (object 0xf1327fb0, a java.util.HashSet),
which is locked by "Finalizer" "Finalizer": waiting to lock
monitor 0xelb70 (object 0xf1327fe8, a weblogic.jdbc.jta.
ResultSet), which is locked by "ExecuteThread: '5' for queue:
'default' "
```
- The problem occurred on a server instance that runs a web application that queries and updates an Oracle database. The code where the deadlock was encountered was part of a common user management interface that gathers user information when the user is logging in. It uses the `oracle.jdbc.xa.client.OracleXADataSource` from a connection pool configured in WebLogic Server.
- Analysis revealed that connection tests were synchronized, causing waits to obtain a lock, even when none was available.
- A code fix solved the problem.
- 
- CR093734      Under WebLogic Server 6.1, an error code was not returned correctly when a client tried to get a connection to an unavailable database using Oracle Thin Driver.
- When using `Driver.connect()` to get the connection directly, `SQLException.getErrorCode()` returned 17002.
  - When using a data source to get a connection from a connection pool, configured with Oracle Thin Driver, whose `TestConnectionsOnReserve` setting was true, `SQLException.getErrorCode()` returned 0.
- Analysis revealed that `weblogic.jdbc.common.internal.ConnectionEnvFactory` swallowed the error code.
- A code fix solved the problem.
- 
- CR094645      When a non-zero starting index was passed to the `getStatementProfiles()` method, the traces returned start from the top of the .tsf file, instead at the specified starting index.
- A code fix solved the problem.
- 
- CR095059      In WebLogic Server 6.1 SP04, the the value returned by the `JDBCStatementProfile.getTimeTaken()` method in the `JDBCStatementProfile` interface was always zero.
- A code fix to `weblogic/jdbc/common/internal/ProfileStorage.java` solved the problem.
-

CR095994	<p><code>weblogic.jdbc.rmi.internal.ConnectionImpl</code> lacked connection leak profiling and connection leak detection logic.</p> <p>Three events can trigger the release of a connection back to the pool.</p> <ol style="list-style-type: none"><li>1. <code>peerGone</code> from the client.</li><li>2. un-referenced call from the <code>BasicServerRef</code> during DGC.</li><li>3. Remote client code properly closes the connection.</li></ol> <p>The first two events should trigger a warning to the user to that there is a potential connection leak.</p> <p>This logic was added to <code>SerialConnection.java</code> and <code>ConnectionImpl.java</code>.</p>
CR096710	<p>Weblogic Server 6.1 SP04 did not throw an ORA-00020 message to the log when a connection pool failed to create with the given initial capacity due to not having sufficient number of processes (<code>max_processes</code>) for the database.</p> <p>The problem was solved by a code change to log the message.</p>
CR096922	<p>Under load conditions, when WebLogic Server was calling <code>ResourceAllocator.markBorrowed()</code> and JMX was calling <code>ConnectionPoolRuntimeMBeanImpl.getConnectionDelayTime()</code>, a deadlock condition resulted.</p> <p>A code fix to <code>ResourceAllocator.java</code> solved the problem.</p>
CR097832	<p>Weblogic Server 6.1 SP04 thin driver, Solaris 8, JDK 1.3.1_04, and Oracle 8.1.7.4., deadlocks occurred when connection refresh is turned on.</p> <p>Customer stack trace:</p> <pre>“ExecuteThread: '35' for queue: 'default'” daemon prio=5 tid=0x45ac68 nid=0x30 waiting for monitor entry [0xce901000..0xce9019d8] at weblogic.jdbc.common.internal.ConnectionEnv.destroy(Connecti onEnv.java:571) at weblogic.common.internal.ResourceAllocator  Both the <code>weblogic.jdbc.common.internal.ConnectionEnv.destroy</code> and <code>weblogic.common.internal.ResourceAllocator.release</code> are synchronized methods ().</pre> <p>The problem was solved with a code fix to <code>weblogic/jdbc/common/internal/ConnectionEnv.java</code>.</p>

---

CR099363

In WebLogic Server 6.1 SP04, under load testing, a connection pool with the refresh minutes set to 15, and `TestConnsOnReserve` and `TestConnsOnRelease` set to false threw the following exception:

```
weblogic.common.ResourceException: No available connections
in pool ODSConnectionPool
```

This problem occurred because when only a few of the connections in the pool are used, all the other connections in the pool are reserved for testing at the expiration of the refresh test minutes. At that point, if client asked for a connection the exception was thrown.

The problem was resolved with a code fix that implements two things:

- When used as-is without any other change, refresh will only reserve and test one connection at a time, and will release it immediately. This addresses the locking-all-connections issue.
- If the customer adds a driver property to the pool definition, `secondsToTrustAnIdlePoolConnection`, with a positive integer value, such as 1, 2, 30 etc., then the pool will avoid testing pool connections that are known to have successfully connected to the DBMS within that period. This will speed up refresh and `testConnsOnReserve`.

---

CR101093

In WebLogic Server 6.1 SP04, when the incorrect password is set in the connection pool properties, the following exception is thrown:

```
<Mar 13, 2003 10:35:45 AM IST> <Info> <JDBC> <Sleeping in
createResource()> <Mar 13, 2003 10:35:46 AM IST> <Info> <JDBC
Pool DemPool> <Pool DemPool is created with initial capacity
0> In the earlier versions of WLS. The exception was more
descriptive. The following exception is thrown when the
incorrect password is specified in sp 3. <Mar 12, 2003 9:41:38
AM IST> <Error> <JDBC> <Cannot startup connection pool "Dpool"
weblogic.common.ResourceException: Could not create pool
connection. The DBMS driver exception was:
java.sql.SQLException: ORA-01017: invalid username/password;
logon denied
```

The problem was resolved with a modification to `weblogic/common/internal/ResourceAllocator.java`.

---

# jDriver

Change Request Number	Description
CR083694	<p>Calling <code>getTimeStamp()</code>, <code>getTime()</code>, <code>getDate()</code> or <code>getJavaDate()</code> on <code>weblogic.db.oci.OciCursor</code> with a null <code>Calendar</code> object, resulted in creation of a new calendar.</p> <p>To improve performance, now a single calendar is created and stored in the cursor the first time one of these methods is called.</p>
CR088387	<p>Using a <code>XADataSource</code> on top of <code>jDriver</code> resulted in shrinking heap size until <code>OutOfMemoryError</code> was encountered.</p> <p>In the <code>weblogic.jdbc.oci.Connection</code> class, LOB fields of result sets in a transaction are registered under a connection through the <code>Connection.addLob()</code> method. The registered LOBs are freed (along with the corresponding object in native <code>jDriver</code> library) when one of these conditions occurs:</p> <ul style="list-style-type: none"> <li>■ <code>Connection.commit()/Connection.rollback()/Connection.close()</code> is called at the connection level (this <code>Connection</code> is <code>weblogic.jdbc.oci.Connection</code>).</li> <li>■ A SQL statement is executed and the connection is set to <code>autoCommit</code> mode (i.e., non-TX Data Source or a direct connection from the pool).</li> <li>■ The return code from an OCI SQL call indicates that a commit/rollback at database level has occurred.</li> </ul> <p>When an <code>XADataSource</code> is being used, none of the above conditions apply. As a result, LOB fields of <code>jDriver</code> in result sets were never released in the JVM heap.</p> <p>This problem was corrected by a code change in <code>weblogic.jdbc.oci.xa.DataSrcThreadInfo.java</code> to call <code>closeLob()</code>.</p>

## 6 Resolved Problems

---

---

CR090025	<p>In WebLogic Server 6.1 SP04 jDriver for Oracle 9.2 does not support the AL32UTF8 character set (unicode version 3.1). When NLS_LANG is set to AMERICAN_AMERICA.AL32UTF8, the following error is generated by weblogic.jdbc.oci.Connection.&lt;init&gt;(Connection.java:246):</p> <pre>java.sql.SQLException: Unsupported Oracle codeset: al32utf8. Set weblogic.codeset in your connection Properties to a valid JDK codeset which is compatible with the Oracle codeset defined in your NLS_LANG setting.</pre> <p>When NLS_LANG=AMERICAN_AMERICA.UTF8, this error occurs: ORA-01461: can bind a LONG value only for insert into a LONG column, indicating a mismatch between the character set on the client and database.</p> <p>This problem was corrected by a code fix to weblogic/db/oci/OciConnection.</p>
CR091151	<p>In WebLogic Server 6.1 SP03 and SP04, jDriver for Oracle ResultSet#getBigDecimal() method did not return correct value. For example, value of 9999999999.999999 was rounded to 9999999999.999998.</p> <p>This problem was solved with a code fix.</p>
CR093795	<p>The “Euro” currency symbol was retrieved as “?” using statement.getString() with WebLogic jDriver for MSSQL Server 2000.</p> <p>The problem was solved by a code fix to use new String constructor to handle character set correctly</p>
CR098071	<p>The version of the Oracle Thin driver 9.2.0 bundled with previous releases of WebLogic Server contained errors that occasionally resulted in data errors. The driver was replaced with a new version from Oracle.</p>
CR101517	<p>Oracle's SQL DATE type supports date values between 4712 B.C. and 4712 A.D., but jDriver for Oracle can not handle dates '1899-12-31' and before. Such dates that are stored by jDriver cannot be queried by other tools, like Oracle Thin Driver or SQL*Plus. This issue happens only if you use a prepared statement, and bind such date to DATE column using setDate() method.</p> <p>The problem was solved by a code fix to weblogic/db/oci/OciColumn.java.</p>

---

CR102060

In WebLogic Server 6.1 SP02, with `jdbcDriver` for Oracle, and Oracle 8.1.7, the `PreparedStatement` for `sql UPDATE` did not work after using `CallableStatement` on the same connection. This was a regression of “CR080931”. `getUpdateCount()` returned zero, and nothing was changed in the database. This occurred when setting the `weblogic.oci.min_bind_size` property for the `jdbc` connection.

```
props.put("weblogic.oci.min_bind_size", "660")
```

Analysis revealed that `jdbcDriver` was using `min_bind_size=33000` for the statement.

```
weblogic.jdbc.oci.Connection#prepareCall internally set  
min_bind_size to 33000.
```

The problem was solved with a code fix to `db\oci\OciConnection.java`.

---

# JMS

---

<b>Change Request Number</b>	<b>Description</b>
CR080296	In WebLogic Server 6.1 SP02, the messaging bridge did not work properly with null <code>UserPassword</code> . When a bridge is configured with a non-null <code>UserName</code> and null <code>UserPassword</code> , the bridge failed to start with no clear error messages.  This problem was resolved with a code fix.

---

CR083933

In WebLogic Server 6.1 SP03., a JMS Server threw a `NullPointerException` under heavy loads:

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <XA resource [JMS_JMSServer1JDBCStore] has not responded in the last 120 second(s).>
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >
```

```
<Aug 2, 2002 6:54:38 PM EDT> <Alert> <JMS> <JMSServer "JMSServer1",unhandled exception during rollback, java.lang.NullPointerException.java.lang.NullPointerExceptio
```

```
n at
weblogic.jms.backend.BEDurableTopicMessageInfo.rollbackReceiveTran(BEDurableTopicMessageInfo.java:352)
at weblogic.jms.backend.BEXATranEntrySubscribe.startRollback(BEXATranEntrySubscribe.java:145)at
weblogic.jms.backend.BEXATranEntry.execute(BEXATranEntry.java:127) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)
at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

The problem was exhibited in this configuration: 400 execute threads, 200 JMS threads, and 100 connections for the connection pool. Exception occurred under a load test while publishing approximately 10 messages (around 2K) to a durable subscriber every second to the JMS Server. There were no distributed transactions. Sybase driver was used for the connection pool for the JDBCStore.

The problem was solved by a code fix to check `prepareStoreRequest` before calling `rollbackReceiveTran()`.

---

---

CR086976	<p>In WebLogic Server 6.1 SP02, JMS JDBS store creation on an AS400 machine with DB2 failed.</p> <p>This problem was solved with a code fix.</p>
CR089114	<p>A JMS selector performance enhancement has been implemented. The customer use case is publishing a message to a small subset of 20,000 subscribers, where each subscriber has a unique identifier. This was CPU-intensive, as it was necessary to evaluate the selector of every subscriber to determine which subset matches.</p> <p>To improve performance under such circumstances, a limited subscriber index capability was added to JMS. With this enhancement, each subscriber informs JMS of its "uniqueness" in a way that JMS knows it can index the subscriber. The uniqueness is specified by using a JMS selector of the form:</p> <pre>xxxxx IS NOT NULL</pre> <p>Where xxxxx is an arbitrary identifier that is not a known JMS message attribute, and where multiple subscribers may specify the same value for "xxxxx". This is standard JMS syntax. WebLogic JMS sees this selector as an indication that it can optimize by indexing the subscriber.</p> <p>When a message is published to a topic which has indexed subscribers, JMS now iterates over each of the message's user properties and passes the message to those indexed subscribers that match the user property name. JMS will also continue old behavior when a message is published, as it will continue to also iterate over unindexed subscribers (those with selectors that don't match the special pattern above).</p>

---

CR089583

In WebLogic Server SP02, a JMSEException occurred when unsubscribing the Durable subscriber:

```
weblogic.jms.common.JMSEException: Subscription clientID.vIJAY
in use, uncommitted/unacknowledged messages at
weblogic.jms.backend.BEConsumer.delete(BEConsumer.java:1784)
at
weblogic.jms.backend.BEManager.removeSubscription(BEManager.
java:204) at
weblogic.jms.backend.BEManager.invoke(BEManager.java:216) at
weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Re
quest.java:510) at
weblogic.jms.dispatcher.DispatcherImpl.dispatchAsync
(DispatcherImpl.java:149) at
weblogic.jms.dispatcher.Request.dispatchAsync(Request.java:7
34) at
weblogic.jms.frontend.FEManager.removeSubscription(FEManager
.java:288) at
weblogic.jms.frontend.FEManager.invoke(FEManager.java:320)
at
weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Re
quest.java:510) at
weblogic.jms.dispatcher.DispatcherImpl.dispatchAsync
(DispatcherImpl.java:149) at
weblogic.jms.dispatcher.DispatcherImpl.dispatchSyncFuture
(DispatcherImpl.java:300) at
weblogic.jms.dispatcher.DispatcherImpl_WLSkel.invoke(Unknown
Source) at
weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.j
ava:298) at
weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServ
erRef.java:267) at
weblogic.rmi.internal.BasicExecuteRequest.execute
(BasicExecuteRequest.java:22) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

Analysis revealed that WebLogic Server was incrementing the unack'ed message count twice for every message and decrementing it only once. This always left the unack'ed messagecount > 0 and thus the exception when we try to unsubscribe the durable subscriber.

This problem was solved with a code fix.

---

CR089682 In WebLogic Server SP03, retry values (min, max and increment) do not work correctly for MessagingBridge. The problem occurred with a MessagingBridge configured to send messages from one WebLogic Server JMS server to another. All run on the same WebLogic Server instance. Untargeting one of the JMS resulted in this error:

```
<30-Oct-02 15:13:34 GMT> <Error> <Connector> <Error granting connection request.>
```

Retry attempts should occur at intervals controlled by the Connection Retry parameters. However, with the following startup default values: min=15, max=60, inc=5. The observed behavior was:

1. A connection is attempted which fails - OK
2. A second connection is attempted 15 seconds later - OK
3. The subsequent connection attempts are at an interval of 5 seconds - NOT OK

```
<30-Oct-02 15:13:09 GMT> <Error> <Connector> <Error granting connection request.>
```

```
<30-Oct-02 15:13:14 GMT> <Error> <Connector> <Error granting connection request.>
```

```
<30-Oct-02 15:13:19 GMT> <Error> <Connector> <Error granting connection request.>
```

```
<30-Oct-02 15:13:24 GMT> <Error> <Connector> <Error granting connection request.>
```

```
<30-Oct-02 15:13:29 GMT> <Error> <Connector> <Error granting connection request.>
```

After one of the default parameters was changed, increments are correct. However, the max value is never reached. With the default values, the maximum reached is 55.

This problem was solved with a code fix to correct the logic for the retry values.

---

CR091195 In WebLogic Server SP04, persistent queue messages received by MDB were not acknowledged properly, causing the Administration Console monitoring function to show "BYTES PENDING". Messages that had been received by the MDB were redelivered when WebLogic Server was shutdown and restarted.

This problem was solved with a code fix MDB acks when the EJB transaction descriptor is NotSupported.

---

CR091827 In WebLogic Server SP02, JDBC logging threw a `java.sql.SQLException: Connection has already been closed with OracleThinDriver for XA, Windows-NT, JDK 1.3.1, and Oracle 9011`. After this error message, nothing got logged in JDBC log file.

This problem was solved with a code fix to prevent JDBC logging errors.

---

CR092464

In WebLogic Server SP02 and SP04, when a durable topic subscriber was connected to a JMS server and there is a message sent to the topic, WebLogic Server increased the value of “Bytes Current Count” by message size even though the message had been received and committed by the receiver. The corresponding value of “Messages Current Count” is not affected. When the durable topic subscriber was not connected to a JMS server and there was a message sent to the topic, WebLogic Server increases both “Messages Current Count” and “Bytes Current Count”. If the durable topic subscriber is connected back to the JMS server, WebLogic Server decreased both “Messages Current Count” and “Bytes Current Count” correspondingly. As a result of this inconsistency, the value “Bytes Current Count” cannot be zero even though there is no outstanding message for the durable topic subscriber.

Analysis revealed that in `backend/BEConsumer.java` `addMessages()`, `bytesCurrentCount` was updated inappropriately, resulting in display of incorrect statistics in the Administration Console.

The problem was solved with a code fix.

---

CR093712

In WebLogic Server SP04, after the maximum idle time out for the messaging bridge, the messages shown below are written to the server log file. If many messaging bridges are configured, the log file grows quickly to a large size.

```
####<Dec 19, 2002 12:44:42 PM EST> <Info> <MessagingBridge>
<dws-stage> <DwsStageServer> <ExecuteThread: '3' for queue:
'MessagingBridge'> <> <> <200027> <Bridge "DWSBackend Inbound
Bridge" works in asynchronous mode and has not received
messages for the predefined maximum idle time. The connections
to the adapters will be interrupted and re-established.>
```

```
####<Dec 19, 2002 12:44:42 PM EST> <Info> <MessaginBgridge>
<dws-stage> <DwsStageServer> <ExecuteThread: '3' for queue:
'MessagingBridge'> <> <> <200020> <Bridge "DWSBackend Inbound
Bridge" is stopped.> ####<Dec 19, 2002 12:44:42 PM EST> <Info>
```

```
<MessagingBridge> <dws-stage> <DwsStageServer>
<ExecuteThread: '3' for queue: 'MessagingBridge'> <> <>
<200033> <Bridge "DWSBackend Inbound Bridge" is getting the
connections to the two adapters.> ####<Dec 19, 2002 12:44:42
PM EST> <Info> <MessagingBridge> <dws-stage> <DwsStageServer>
<ExecuteThread: '3' for queue: 'MessagingBridge'> <> <>
<200032> <Bridge "DWSBackend Inbound Bridge" is configured not
to allow degradation of its quality of service in cases where
the configured quality of service is not reachable.>
```

```
####<Dec 19, 2002 12:44:42 PM EST> <Info> <MessagingBridge>
<dws-stage> <DwsStageServer> <ExecuteThread: '3' for queue:
'MessagingBridge'> <> <> <200030> <Bridge "DWSBackend Inbound
Bridge" is configured to work in "Exactly-once" mode and it is
actually working in "Exactly-once" mode.> ####<Dec 19, 2002
12:44:42 PM EST> <Info> <MessagingBridge> <dws-stage>
<DwsStageServer> <ExecuteThread: '3' for queue:
'MessagingBridge'> <> <> <200028> <Bridge "DWSBackend Inbound
Bridge" starts transferring messages.>
```

This problem was resolved by providing the ability to hide the log messages with `DebugMessagingBridgeRuntime` debug flag.

---

CR099455

JMS messages were getting lost due to race conditions.

This problem was corrected with a code fix.

---

CR101374      JMS messages were lost. A JMS Server had an inbound and an outbound queue. An MDB located on another server consumed messages from the inbound queue and produced messages to the outbound queue in a one-for-one relationship. This was done within a transaction initiated by the MDB. The end of the transaction is when the MDB enqueued a message on the outbound queue. If the JMS Server was killed and restarted, messages were lost. Over a load of 10,000 messages, a small number of messages were lost.

The problem was solved by a code change to force transacted `XACONNECTION` to `CLIENT_ACK`.

---

CR101670      During tests, a JMS client session that never committed or acknowledged leaked memory. The test used startup classes that act as Asynchronous receivers for both queues and topics. The receiver was created using a transacted session and `AUTO-ACKNOWLEDGE` mode. On receipt of messages, the receiver did a `Rollback` on the session. Diagnostics revealed a memory leak in `weblogic.jms.client.JMSSESSION$UnackedMessage`.

The problem was solved with a code fix to `weblogic/jms/client/JMSSESSION.java`.

---

## JNDI

---

<b>Change Request Number</b>	<b>Description</b>
CR087237	<p>In WebLogic Server 6.1 SP03, in a two node cluster running under Solaris, failover of a stateless session bean was delayed by about five minutes. The server froze and the client hung.</p> <p>The time to failover was reduced to 45 seconds with multiple code and configuration changes:</p> <ul style="list-style-type: none"><li>■ <code>RJVMFinder</code> was made as <code>PeerGoneListener</code> to remember the unavailable server and not try the server until the replica list is empty.</li><li>■ Fix to <code>BasicReplicaHandler</code> to avoid <code>refreshReplicaList</code> calls during failover, when it finds the replica list has more servers.</li><li>■ <code>T3JVMConnection</code> has new functionality, to create the socket on a worker thread. In addition, the same timeout is used for the <code>so_timeout</code>, so on solaris8 clients, read will not block for one minute. The new flag is <code>-Dweblogic.client.SocketConnectTimeoutInSecs=15</code> flag.</li><li>■ Control heartbeat period on server and client using the existing <code>-D</code> values. Increase the default execute queue on server to take care of extra heartbeat load. Client heartbeat time is controlled by <code>-Dweblogic.PeriodLength=10000</code> flag. Server heartbeat is controlled by <code>PeriodLength="10000"</code> flag in the <code>ServerMBean</code> in the <code>config.xml</code>.</li></ul>
CR093700	<p>When (re)binding and unbinding clusterable RMI objects from JNDI tree, the unbind operation was sometimes making the object null, setting it up for garbage collection, but still holding a reference to the null object when the object was hosted locally on the server.</p> <p>A code fix was implemented to ensure that the object is set for garbage collection only when no other replica of this object is available in the cluster.</p>

---

---

CR093799	<p>When performing a <code>Context.lookup</code> on a composite name, a <code>NameNotFoundException</code> occurs if the name was not resolved, either fully or partially.</p> <p><code>Name remainingName = nnfe.getRemainingName();</code> should obtain the components that were not resolved from the name. These are the subcontexts that need to be created in the JNDI tree.</p> <p>So when enumeration <code>e = remainingName.getAll()</code> is called, the enumeration obtained should contain the unresolved components as string elements. e.g ("Unresolved1", "Unresolved2", "Unresolved3"). However, the Enumeration contains only one string, "Unresolved1.Unresolved2.Unresolved3"</p> <p>The problem was solved by a change to <code>BasicNamingNode.java</code> to ensure that <code>CompositeName</code> is constructed with the '/' separator.</p>
CR096838	<p>In WebLogic Server 6.1 SP03 and SP04 <code>weblogic.jndi.Environment</code> objects were not released on a timely basis, causing an <code>OutOfMemoryError</code> error during lookups for <code>ejb-ref-name</code> on the <code>InitialContext</code>.</p> <p>A code fix was implemented to correctly close the context.</p>

---

## JSP

Change Request Number	Description
CR092039	<p>WebLogic Server 6.1 SP03, the <code>JspParser</code> threw an <code>UnsupportedEncodingException</code> for the extra quoted value in charset with JSP tag:</p> <pre>&lt;%@ page contentType="text/html; charset=\"Shift_JIS\" " %&gt;</pre> <p>resulted in this error:</p> <pre>java.lang.RuntimeException: Unknown/unsupported charset: "Shift_JIS" - java.io.UnsupportedEncodingException: Charset: '"Shift_JIS"' not recognized, and there is no alias for it in the WebServerMBean</pre> <p>WebLogic Server did not correctly support the HTTP specification.</p> <p>The problem was resolved by adding support for quoted strings and comments to <code>Content-Type</code> header charset attribute value.</p>
CR088019	<p>In WebLogic Server 6.1 SP03, a generated java file failed to compile on HP-UX 11 with virtual host. The <code>.ear</code> file deployed correctly when not targeted to a virtual host. When accessed, <code>login.jsp</code> generated the java file and compiled it.</p> <p>A virtual host was configured, and the webapp part of the <code>.ear</code> was targeted to the virtual host.</p> <p>After restart of the server, when accessed, <code>login.jsp</code> generated the java file but failed to compile it into a class file.</p> <p>The problem was solved with a code fix. The temporary directory now contains an element identifying the <code>WebServer</code> component so as to distinguish between virtual hosts and/or the default web-server. This prevents deployment file clashes when WebApps are deployed and the temporary directory is cleaned out.</p>
CR092089	<p>A JSP file with multibyte name did not compile, and the directory returned its contents. Analysis revealed that the JSP file path was not correctly detected when the file name contained multibyte characters. WebLogic Server decoded the path by VM default, then the path was decoded again using 'input-charset'.</p> <p>Browser decoding implementation was a related issue. Internet Explorer performs <code>URLEncode</code> based on UTF-8. Netscape does it based on platform default.</p> <p>The problem was resolved by a code fix to decode <code>ContextPath</code>, <code>ServletPath</code>, and <code>PathInfo</code> based on UTF-8.</p> <p>Decoding method can be configured with <code>weblogic.http.URIDecodeEncoding</code>.</p>

CR092366	<p>The <code>javax.servlet.jsp.PageContext.out</code> attribute was not cleaned up correctly for body content tags.</p> <p>The problem was solved with a code fix.</p>
CR093014	<p>Java comments that start in one scriptlet and end in the next one cause an error:</p> <pre>&lt;22.07.2002 14:12:24 CEST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;[WebAppServletContext(7422744,DefaultWebApp,/DefaultWebApp) ] Servlet failed with Exception weblogic.servlet.jsp.JspException: (line 12): scriptlet close brace '}' unbalanced at line 12 which breaks scope '_base_service_scope_' at ...</pre> <p>The problem was solved by back-porting grammar from WebLogic Server 7.0 that caused the <code>ScriptletScopeLexer</code> to skip an entire Java comment.</p>
CR093291	<p>JSP compile by the server did not include the system classpath. On server startup, JSPs that referenced a jar in CLASSPATH did not compile.</p> <p>This message occurred:</p> <pre>package com.bea.p13n.util.debug does not exist probably occurred due to an error in /devtools/debug.jsp line 1:&lt;%@ page import="com.bea.p13n.util.debug.Debug" %&gt;</pre> <p>Analysis revealed that the problem occurred when the jar in system classpath began with <code>"/.."</code>.</p> <p>The problem was solved with a code fix to support consider <code>./</code> and <code>../</code> cases when cleaning the classpath.</p>
CR093625	<p>In WebLogic Server 6.1 SP03 and SP04, server misdirected output for JSP includes in JSP pages. The problem was solved with changes to the servlet engine to unify handling of includes, forwards, wrapped-responses, and JSP BodyTags.</p>
CR094190	<p>According to the JVM specification, the size limit for a method size is 64K. In WebLogic Server 6.1 SP03, a customer has JSPs with many body tags for which the generated java code exceeded the 64K limit for the <code>__jspService</code> method.</p> <p>The problem was solved by a code change. Now, when an empty BodyTag of the form <code>&lt;my:tag ... /&gt;</code> is encountered, the generated code that would normally set up scope for the body is replaced by a single call to a static method in <code>StandardTagLib.java</code> that tricks the BodyTag into thinking it is being invoked as usual from the JSP source.</p>
CR098402	<p>In WebLogic Server 6.1 SP04, <code>jsp:setProperty</code> converted a request-time expression in the property value to String. The problem was solved with a code fix.</p> <p>When assigning from a value given as a request-time attribute, no type conversions are applied.</p>

# JTA

---

<b>Change Request Number</b>	<b>Description</b>
CR088171	<p>In WebLogic Server 6.1SP02 when a transaction is in unknown state the transaction is lost</p> <p>Given the following cluster configuration:</p> <p>MS-1</p> <ul style="list-style-type: none"><li>■ Queue1 - MDB1 TX Required</li><li>■ Queue2 - MDB2 TX Required</li><li>■ SLSB</li></ul> <p>MS-2</p> <ul style="list-style-type: none"><li>■ Queue3 - MDB3 TX required</li><li>■ SLSB - <code>send()</code> - TX required</li><li>■ <code>sendImmediate()</code> - TX RequiresNew</li></ul> <p>The queues are persisted to a DB, MDBs transactionally de-queueing messages. The following occurs:</p> <ol style="list-style-type: none"><li>1. MDB3 -&gt; <code>SLSB.send()</code> and then <code>SLSB.sendImmediate()</code></li><li>2. the SLSB.send does: <code>SLSB.send() ...&gt;MDB2-&gt; MDB1 -&gt; SLSB.sendImmediate()</code> called twice</li><li>3. Send 100 messages to Queue3, allow processing to commence. Kill MS-1 after all messages have been enqueued but prior to all the messages being consumed from Queue3.</li><li>4. Restart MS-1.</li></ol> <p>When MS-2 is killed, the MS-1 receives a <code>peergone</code> exception and the transaction which is ready to commit its status goes from prepared to unknown. When the second server is brought back up the transaction is lost because it does not know what to do is commit or rollback because of the UNKNOWN state.</p> <p>The problem was solved with a code fix to rollback a transaction when an unknown exception is received during commit.</p>

---

CR088844      In WebLogic Server 6.1 SP02, a deadlock between LogManager and JTA occurred on server startup. Thread dump analysis revealed:

Execute thread 7: This thread is trying to enlist a resource. It calls `ResourceDescriptor.startResourceUse`. Inside the synchronized block of this method is a trace statement.

Execute Thread 6: The Audit Bean is trying to make a call to `LogBroadcaster`. This makes an RMI call that causes the transaction to be suspended and delist the resources. Delist makes also a call on `ResourceDescriptor.startResourceUse`.

The trace statement inside the `startResourceUse` in thread 7 is blocking waiting on the `LogBroadcast` in thread 6 and the enlist of thread 7 is waiting for the delist in thread 6.

The problem was solved by moving the trace statement in `startResourceUse` outside the synchronized block.

---

CR091628      The `javax.transaction.xa.XAResource` interface defines a method, `setTransactionTimeout()`, that allows the transaction manager to set the transaction timeout in the resource manager for operations associated with the `XAResource` instance. The WebLogic Server transaction manager did not invoke this method when a resource manager is accessed by an application in a global transaction. This problem was solved by adding a `XATransactionTimeout` property to be used when `XASetTransactionTimeout` is "true". This is because, if the JTA Transaction Timeout value is used and that value is high, if WebLogic Server crashes in mid-transaction, Oracle locks the rows and no one can access them unless Oracle is cycled.

---

CR091882      In WebLogic Server 6.1SP03 a transaction timeout occurred when calling a stateless EJB in a remote 6.1SP03 Cluster.

The problem was solved by a code fix to `transaction/internal/TransactionImpl.java`.

---

CR092301      In WebLogic Server 6.1 SP04, a null point exception was thrown by `weblogic.transaction.internal.ServerResourceInfo.isAccessibleAtAndAssignableTo`. The problem occurred intermittently even when the server is idle, and could not be reliably reproduced.

The problem was resolved with a code fix to `weblogic/transaction/internal/ServerResourceInfo.java`.

---

CR093045

In WebLogic Server 6.1 SP02 and later, the Oracle 8.1.7.4 thin driver XA connection pool became unusable after the database stopped and started under WebLogic Server load.

The test application used XA connection pool to update columns in Oracle 8.1.7.4 via MDBs. When WebLogic Server is processing the MDB messages and the Oracle database is stopped/restarted the XA connections are not refreshed and distributed locks in Oracle are not recovered (ORA-01591). WebLogic Server goes into a cyclic “hang” situation and tries to refresh the XA connections. WebLogic Server log file shows following when “cycling” to refresh/use XA connections:

```
... #####<13.12.2002 12:09:54 CET> <Info> <JDBC Pool
FoundationThinXAPool> <A4PMA9ALX> <myserver> <ExecuteThread:
'9' for queue: 'default'> <guest> <484:6c709abb7f28edb9>
<000000> <javax.transaction.SystemException: start() failed
on resource 'weblogic.jdbc.jta.VendorXAResource' null at
weblogic.transaction.internal.ServerResourceInfo.xaStart(Ser
verResourceInfo.java:1010) at ...
```

WebLogic Server never recovered from this situation (cycles). All threads of the server seemed blocked and server had to be killed (`weblogic.Admin SHUTDOWN` not working). Only restarting the WebLogic Server resolved the distributed locks in Oracle as soon as JTA recovery takes place. Before the recovery is done many ORA-01591 exceptions were thrown as the MDBs try to re-deliver their outstanding messages. After JTA recovered the in-doubt distributed transactions, processing of outstanding JMS messages completes successfully.

The problem was solved with code fixes to  
`weblogic\jdbc\oci\xa\XADataSource.java`.

---

CR100357

When servers participating in a transaction were killed and restarted, recovery was slow, or deadlock occurred. The configuration included four JMSServers with pinned connection factories and distributed inbound and outbound queues, four MDB Servers (each with four MDBs deployed, five beans in pool listening to each distributed member in the inbound queue), 40,000 messages of size 2K in the inbound queue.

When JMS and MDB servers were restarted, the JMS servers used up all default queue threads and the Execute Queue continued to grow.

`ServerTransactionImpl.java` was modified to fix the problem.

---

---

CR100898

If a transaction was rolled back, and there was a server participant that had no resource participants, it was possible that the transaction would never be completed. The transaction would continue to exist on the coordinating server and continually retry notifying the server with no resources to roll back. Eventually, the transaction would be abandoned. Although all resource participants were informed of the rollback, it was possible that after completion callbacks would not be processed on the coordinating server.

This problem was resolved by logic changes in rollback async reply processing to accommodate servers with no resources and to immediately set state to rolled back when all servers/resources are accounted for.

---

### Miscellaneous

---

<b>Change Request Number</b>	<b>Description</b>
CR094764	<p>In WebLogic Server 6.1 SP04, <code>beasvc.exe</code> did not shut down gracefully when Windows-NT was restarted. The problem occurred when the user:</p> <ol style="list-style-type: none"><li>1. Configured WebLogic Server to run as NT service.</li><li>2. Started WebLogic Server from the Windows Control Panel.</li><li>3. Restarted Windows-NT from the Windows Start menu.</li></ol> <p>The server log was empty, and the event viewer contained this message:</p> <pre>The myserver service terminated unexpectedly. It has done this 1 time(s).</pre> <p>The following corrective action will be taken in 0 milliseconds: No action.</p> <p>Analysis revealed that the Windows Service Control Manager (SCM) should be notified of <code>SERVICE_ACCEPT_SHUTDOWN</code> so that it will inform <code>beasvc.exe</code> of <code>SERVICE_CONTROL_SHUTDOWN</code> during system shutdown.</p> <p>A code change to implement the required processing solved the problem.</p>
CR099590	<p>In WebLogic Server 6.1 SP02, using a JMX client to delete a managed server resulted in a null pointer exception at</p> <pre>weblogic.management.internal.MBeanHomeImpl.getInterface(MBeanHomeImpl.java:431)</pre> <p>The problem was solved with a correction to the mbean delete.</p>

---

### Plug-ins

The plug-in(s) in which the problem was resolved is shown in parentheses in the Description column.

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

091740 (HttpClusterServlet) During benchmark tests, 503 and 500 HTTP error were being returned by the proxy server running the HttpClusterServlet. The errors were random and occasional. The configuration was JRockit 7.0 SP2 Load 2 and the Sun JDK 1.3.1\_06. AcceptBacklog and the thread count were increased to 1000 and 50 respectively. The benchmark was run with 100-200 client threads.

This problem occurred when the servlet's dynamic server list contained no entries. The problem was solved with a code fix to check the size of the dynamic server list of servers before creating an array out of that list.

---

CR084625 (NSAPI) All version of the iPlanet plugin had `/_wl_proxy/_post` hardcoded:

```
#else /* UNIX */
sprintf(name, "/tmp/_wl_proxy/_post__%d_%d", pid,
postCounter);
fsep = '/';
#endif
```

This poses problems for customers that have multiple servers using different user IDs and groups.

For example, uploaded files from clients are saved temporarily in `/tmp/_wl_proxy` of web servers that proxy to WebLogic Server. The `_wl_proxy` directory is removed each time the web server is rebooted. If the environment is shared (multiple instances on the same physical servers) the user process that creates this directory blocks other users from writing to it, because it uses its default umask to create the permissions (i.e., application A [running as User ID A] creates `/tmp/_wl_proxy/<some temp file>` and then application B tries to upload a file and gets the following error in the error file:

```
[27/Aug/2002082140] failure (14367) for host 172.18.5.122
trying to POST /filetransfer/TransferUtility.do, wl-proxy
reports exception occurred 'READ_ERROR [os error=13, line 501
of proxy.cpp] Cannot open TEMP file
'/tmp/_wl_proxy/_post__14367_5' in $TMP/_wl_proxy for POST of
2867 bytes
```

This problem was solved by adding a configurable `WLTempDir` argument. This argument applies to both `wlproxy.log` and `_wl_proxy`. For backward compatibility, `WLogFile` overrides `WLTempDir` during log file creation.

---

---

CR085192	<p>(NSAPI and ISAPI) Performance degradation occurred when connection pooling was turned off.</p> <p>Resource requirements for URL creation were high.</p> <p>WebLogic closed the connection before the plug-in did, resulting in “half-open” connections, and <code>PROTOCOL_EXCEPTIONS</code> in the plug-in.</p> <p>The problems were addressed by multiple code changes:</p> <ul style="list-style-type: none"><li>■ <code>Keep-Alive</code> is enabled by default for ISAPI and NSAPI.</li><li>■ WebLogic Server now sends the <code>keepAliveSecs</code> value to the plug-in in a special header.</li></ul>
CR085285	<p>(NSAPI) The web application container returned 404 messages for <code>InitJVMID</code> requests from the plugins, resulting in log files filling up with messages:</p> <pre>#####&lt;Sep 6, 2002 4:56:43 PM EDT&gt; &lt;Debug&gt; &lt;HTTP&gt; &lt;petunia&gt; &lt;mslpetunia&gt; &lt;ExecuteThread: '13' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;101147&gt; &lt;HttpServer(887891,null default ctx,mslpetunia) found no context for "/WLDummyInitJVMIDS".</pre> <p>The problem was solved by sending <code>InitJVMID</code> requests to the internal web application which is always deployed on WebLogic Server.</p>
CR085541	<p>(all plug-ins) <code>__WebLogicBridgeConfig</code> returned internal representation of <code>Debug</code>—a confusing 10 character code.</p> <p>A code fix was made to return the value represented by the code.</p>
CR085695	<p>(ISAPI) When <code>KeepAlive</code> was enabled, the time to close the connection is about 30 seconds with a <code>HEAD</code> HTTP request.</p> <p>This problem was exhibited under 6.1 SP01, Windows 2000, IIS 5.0.</p> <p>This problem was corrected with a code fix to close the connection immediately for <code>HEAD</code> request.</p>
CR085921	<p>(ISAPI) For all plug-ins, it was not possible to differentiate between <code>WRITE_ERRORS</code> that occurred while writing to the request/post-data to WebLogic Server and <code>WRITE_ERRORS</code> that occurred while writing response headers/data to the browser in the <code>__WebLogicBridgeConfig</code> statistics. This deficiency made it more difficult to use the statistics displayed by <code>__WebLogicBridgeConfig</code> for plug-in/cluster health monitoring.</p> <p>This deficiency was corrected by implementing two new error types <code>WRITE_ERROR_TO_CLIENT</code> and <code>WRITE_ERROR_TO_SERVER</code>.</p>

---

---

CR085923 (ISAPI) For all plug-ins, setting Debug=ERR caused INF messages to also be logged. This problem was corrected by a code fix. Now, only ERR messages are logged when Debug=ERR.

---

CR086224 (NSAPI) Plug-in did not read the SESSIONID from post data. Analysis revealed a problem in `getPreferredServersFromCookie()`. When `session.getId()` was kept as post data, `getId()` included an extra string: `|time`. The problem was solved by a code fix.

---

CR086490 (ISAPI) For all plug-ins and `HttpClusterServlet`, the `MaxSkips` feature was disabled, because its default value of 10 proved to make the feature less useful than desired. The lack of this feature posed potential performance degradation when one or more servers in the cluster are unresponsive. This problem was resolved by providing a new parameter, `MaxSkipTime`. `MaxSkipTime` can be used to specify the time in seconds to wait after marking a server bad until forcing a new server list.

---

CR086518 (NSAPI) If performing a redirect using a URL that contains both a query string and an anchor, the anchor was incorrectly considered part of the value portion of the last query string member. This problem only occurred when using Internet Explorer in a three-tier environment, using iPlanet 6.0 for the HTTP server). Using Netscape or Mozilla, or if running the server in a two-tier environment, the problem did not occur.

To replicate, invoke `foo.jsp`.

```
foo.jsp:
  <%
    String redirectURL =
"http://hostname/T/foo2.jsp?a=b#foo"; ;
    response.sendRedirect(redirectURL);
  %>
```

```
foo2.jsp:
<%= request.getParameter("a") %>
```

The Administration Console displayed the value for "a" as "b#foo", instead of "b".

The problem was solved by a code fix to trim the anchor from the query string, because Internet Explorer does not do so for `sendRedirected URL`

---

CR088914 (NSAPI) See "CR088915" on page 6-202

---

CR088915

CR088914

(ISAPI, NSAPI) A change was made to reduce vulnerability to SSL certificate chain attacks. Code to check `BasicConstraints` for CA certificates was added. Constraint checking is controlled with the plug-in parameter `EnforceBasicConstraints`. The parameter settings are now:

- **OFF**—disables the enforcement entirely is not recommended as a solution unless you really have no other choice. For example, a customer has bought certificates from a commercial CA and the chain doesn't pass the new check. Note that most current commercial CA certificates should work under the default **STRONG** setting.
- **STRONG**—`BasicConstraints` for V3 CA certificates are checked and the certificates are verified to be CA certificates. This is the default.
- **STRICT**—This level does the same checking as the **STRONG** level, but in addition it also strictly enforces IETF RFC 2459 which specifies the `BasicConstraints` for CA certificates also must be marked as "critical". Set this if you want to strict conformance to RFC 2459.

This change may result in security exceptions that did not occur in WebLogic Server 6.1 SP04. For example, an attempt to make an outbound SSL call, resulted in this error when the certificate presented by the remote server instance did not have a basic constraint:

```
<May 8, 2003 4:36:18 PM MDT> <Notice> <WebLogicServer>  
<SSLListenThread listening on port 7002> <May 8, 2003 4:36:19  
PM MDT> <Notice> <Management> <Starting discovery of Managed  
Server... This feature is on by default, you may turn this off  
by passing -Dweblogic. <May 8, 2003 4:36:19 PM MDT> <Notice>  
<WebLogicServer> <Started WebLogic Admin Server "myserver" for  
domain "412253" running in Production Mode> EXCEPTION: Socket  
Closed java.io.IOException: Socket Closed at  
weblogic.security.SSL.SSLSocket.sendAlert(SSLSocket.java:108  
6) at...
```

Such errors may be prevented by setting the plug-in parameter `EnforceBasicConstraints` to false. You can set disable checking of basic constraints at the command line with this command:

```
-Dweblogic.security.SSL.enforceConstraints=false
```

---

CR089746

(NSAPI) WebLogic Server supported the following scenarios:

- plug-in via T3s to application server. (No SSL Session caching and NO Keep-Alive)
- plug-in via HTTPS to application server. (No SSL Session caching and NO Keep-Alive) Customer is asking for SSL session resumption between plug-in and Weblogic Server.

A customer requested support for SSL session caching and keep-alives between the plug-in and and WebLogic Server, for T3s and HTTPS.

The request was satisfied by adding the `KeepAliveEnabled` flag , and adding SSL session cache.

---

CR091635 (NSAPI) The SP03 version of the plug-in failed to parse the the JVMIDs for the following request:  
2002 Found Encoded  
Session=[JSESSIONID=9dq5Z081!805492834!174332983!7001!7002?\_JASPER=1000/r/1037920979629&PAGE=HOME\_TEMPLATE&INTEGRATION=YES]  
Thu Nov 21 15:16:09 2002 Parsing cookie  
JSESSIONID=9dq5Z081!805492834!174332983!7001!7002?\_JASPER=1000/r/1037920979629&PAGE=HOME\_TEMPLATE&INTEGRATION=YES  
Thu Nov 21 15:16:09 2002 getpreferredServersFromCookie:  
805492834!174332983!7001!7002?\_JASPER=1000/r/1037920979629&PAGE=HOME\_TEMPLATE&INTEGRATION=YES  
Thu Nov 21 15:16:15 2002 parseJVMID: could not resolve hostname 'r', treating it as invalid  
The problem was solved with a code fix to check JVMID portion of session cookie contains '/' instead of the whole sessionid.

---

CR092756 (ISAPI) ISAPI plug-in was not failing over on receipt of a HTTP 503 response from the backend. A code fix was implemented to solve this problem.

---

CR093196 (NSAPI) JSP `sendRedirect()` failed when web application is deployed on a cluster and the JSP is accessed through the iPlanet proxy plug-in.  
An IE browser returns the message The page cannot be displayed... and Netscape returns The document contained no data...  
The problem was solved with a code fix to check for null value of secondary session.

---

CR093530 (ISAPI) The `PathTrim` value was case-sensitive, causing problems for environments that prefer to allow clients to access an application with a case-insensitive URL.  
This problem was resolved by changing the `PathTrim` value to be case-insensitive.

---

CR093662 (NSAPI) A call to `encodeRedirectURL()` failed when:  
(a) A webapp is deployed as a .war file on a cluster;  
(b) A JSP in the webapp is accessed through the iPlanet proxy plug-in; and  
(c) The JSP invokes `encodeRedirectURL()` in a FORM using the HTTP GET method  
A core dump occurs on the iPlanet Web Server. An Internet Explorer browser returns The page cannot be displayed...Cannot find server or DNS Error and Netscape returns The document contained no data... 1.  
Analysis revealed that the problem was caused by storing primary or secondary jvmid in a fixed length buffer without trimming the query string.  
The problem was solved by a code change to should trim the query string before parsing the session cookie

---

CR094109	<p>(Apache) Failover from primary server did not occur when response time exceeded the time limit set by <code>HungServerRecoverSecs</code>, and the <code>Idempotent</code> primitive is on.</p> <p>Analysis revealed that the problem was related to an error in the exception handling source code in <code>ap_proxy.cpp</code>.</p> <p>The problem was solved by rewriting the failover code.</p>
CR094282	<p>(NSAPI) If the server listed in <code>WebLogicHost</code> and <code>WebLogicPort</code> or in <code>WebLogicCluster</code> was not up and running, or for example, Solaris started in single user mode, the plug-in hung until the operating system timed out the connection.</p> <p>By default, this timeout is lengthy.</p> <p>A new introduce plug-in parameter, <code>WLSocketTimeoutSecs</code>, was added. Use of this parameter prevents long delays when a machine is down. This default value is 2 seconds. The value must be greater than 0.</p>
CR094663	<p>Same problem and resolution as “CR095166” on page 6-205.</p>
CR094768	<p>(ISAPI) Parsing post data for cookies not implemented correctly in plugins. This error resulted</p> <pre>"Bad request in the post data [NameField=aaaaa&amp;ValueField=11111111111&amp;AddValue=+Add+]"</pre> <p>The <code>UrlUnescape()</code> method was returning incorrect return type, and parsing post data for content-types other than <code>application/x-www-form-urlencoded</code>.</p> <p>The bug was corrected with a code fix.</p>
CR095009	<p>(ISAPI) The plug-in could not successfully resolve to the appropriate server instance. The error in the log files was: <code>Primary JVMID not found in the server list, ignore preferred servers</code>.</p> <p>Analysis revealed an error in <code>getPreferredServersFromCookie()</code>. The problem was solved with a code fix.</p>
CR095558	<p>(ISAPI) The plug-in did not differentiate properly between read failures from the browser and write failures to the server. A code change was made to process connection aborted messages (10053/10054) only when failure is between the client and the plug-in.</p>
CR095559	<p>(Apache) After failover of primary server, the plug-in continued to search for the primary server instance, instead of search for the secondary server instance and sending the it the request upon which the primary server failed.</p> <p>The problem was solved by a rewrite of the failover code.</p>

CR095166	<p>(ISAPI) The IIS plug-in parsed cookies incorrectly when the client sent multiple cookies, resulting in the plug-in not sticking to the primary server instance.</p> <p>Analysis revealed an error in the string checking logic, <code>strstr()</code> did not correctly parse the session cookie name.</p>
CR096625	<p>(Apache) The <code>X_WEBLOGIC_FORCE_JVMID</code> header was sent to a server instance in a cluster.</p> <p><code>X_WEBLOGIC_FORCE_JVMID</code> should only be sent when the server list is non-clustered and the current server does not have a <code>jvmid</code> yet.</p> <p>This problem was solved by a code fix.</p>
CR097202	<p>(All plug-ins) A customer need to determine a browser's cipher strength on the WebLogic side, in a configuration that included a plug-in with 128 bit step up certs between WebLogic Server and the browser. The browser had 40 bit or 128 bit strength. The customer used this code to obtain the browsers cipher strength:</p> <pre>(Integer)httpRequest.getAttribute("javax.servlet.request.key-size")</pre> <p>to get the browser's strength.</p> <p>The value returned was always 128, although the browser was set to use a cipher size less than 128 bit. The <code>https_keysize</code> determines the strength of the connection that is made. Because 128 bit certs were used on plug-in, 128 was returned, regardless of the strength used by the browser. When the request was directed to WLS, 40 or 128 was returned, depending on the browser strength.</p> <p>The plug-in does not modify the <code>HTTPS_Keysize</code> or <code>HTTPS_SECRETKEYSIZE</code> headers.</p> <p>To meet this need, to new headers were implemented: <code>WL-Proxy-Client-Keysize</code> and <code>WL-Proxy-Client-Secretkeysizes</code>. Both headers values can be obtained with <code>request.getHeader()</code>.</p>
CR096850	<p>(NSAPI) The iPlanet server, running under Windows2000, crashed when a load testing application is trying to make multiple connections to the server. The iPlanet server crashes after multiple attempts are made to connect to the server.</p> <p>Analysis revealed the iPlanet server crashed because of access violation and it points to <code>MSVCRT!CxxThrowException</code>.</p> <p>The problem was solved by a code fix to <code>sendResponse</code>, to return and handle the exception, instead of throwing the exception.</p>

---

## 6 Resolved Problems

---

---

CR100361	<p>The following new exception types were added:</p> <p>READ_ERROR_FROM_CLIENT—error when reading data from client</p> <p>READ_ERROR_FROM_SERVER—error when reading data from backend server</p> <p>READ_ERROR_FROM_FILE—error when reading from temporary file which stores post data</p> <p>WRITE_ERROR_TO_FILE—error when writing post data to the temporary file</p>
CR101222	<p>(NSAPI and Apache) Failover to a secondary server instance when the primary server instance hung did not occur reliably. After establishing a primary and a secondary session, the client sent a recovery request that slept for longer than the value of <code>HungServerRecoverySecs</code> if the <code>request.getParameter("primary")</code> was equal to the server name. The web server did not detect that the primary server was hung and fail over the request to the secondary.</p> <p>Analysis revealed that the secondary server was marked bad, which caused the plug-in to skip it, and use the general server list.</p> <p>The problem was solved by a code change to reset the server to good after <code>MaxSkipTime</code>.</p>
CR101428	<p>(ISAPI) When requests contained multiple cookies, if cookies not named JSESSIONID were after the JSESSIONID cookie, they were stripped off and not delivered to WebLogic Server.</p> <p>The problem was solved by a code change.</p>
CR101596	<p>(NSAPI) In WebLogic Server 6.1 SP04, <code>ssl_certchain_verify_callback()</code> incorrectly returned <code>SSLNoErr</code> for certain error conditions.</p> <p>The problem was solved by a code change to return <code>X509CertChainInvalidErr</code> when the error conditions are encountered.</p>
CR103126	<p>(Apache) When starting Apache using the plugin on Linux for <code>ssl128</code>, this error occurred:</p> <pre>Cannot load /usr/local/apache/libexec/mod_wl_ssl128.so into server: /usr/local/apache/libexec/mod_wl_ssl128.so: undefined symbol: pthread_mutexattr_init</pre> <p>Analysis revealed a build problem, which was solved by rebuilding the plug-in.</p>
CR103161	Duplicate of " <a href="#">CR097202</a> ".

---

## RMI-IIOP

Change Request Number	Description
CR090010	<p>Under WebLogic Server 6.1 SP03, <code>BoxedRMI</code> IDL file <code>seq2_seq1_octet.idl</code> for multi dimensional arrays, generated with <code>weblogic.ejbcc</code> contained module/path declaration <code>org.omg.boxedRMI</code> twice.</p> <p>The problem was solved with a code fix to correct the IDL declaration for multidimensional <code>boxedRMI</code> sequences.</p>
CR093988	<p>Under WebLogic Server 6.1 SP04, accessing an EJB over IIOP from an ActiveX component caused WebLogic Server to hangs. The thread dump indicated a dead lock in the java layer.</p> <p>The problem occurred with <code>SUN JDK 1.3.1</code>, under <code>Red Hat Linux 7.2</code> and <code>Windows2000</code>.</p> <p>Analysis revealed that IIOP socket timeout trigger closing sockets could cause deadlocks.</p> <p>A code fix was implemented to prevent this scenario.</p>
CR100334	<p>In previous releases of WebLogic Server 6.1 dispatching methods with no arguments or invoking methods with a void return type would cause WebLogic Server to try and read data that potentially was not there.</p> <p>The data is not there when delivered from 7.0sp2, for other releases we try and accommodate 6.1 by delivering the extra (unnecessary) data.</p>
CR100430	<p>WebLogic Server 6.1 SP04 and earlier does not interoperate with WebLogic Server 8.1. Transactions between 6.1 and 8.1 server instances timeout with an <code>ArrayIndexOutOfBoundsException</code> on the 6.1 server instance.</p> <p>This problem is solved in WebLogic Server 6.1 SP05.</p>
CR100334	<p>This release solves an interoperability problem between WebLogic Server 6.1 SP04 and WebLogic Server 7.0 SP02.</p> <p>WebLogic Server 6.1 SP04 erroneously tried to read void parameters that were returned to it, resulting in an <code>UnmarshalException</code>. This occurred when:</p> <ul style="list-style-type: none"> <li>■ WebLogic Server 6.1 SP04 received a void return value as a result of a call to a WebLogic Server 7.0 SP02 server instance</li> <li>■ A method with no arguments was invoked on a 6.1 server instance by a WebLogic Server 7.0 SP02 server instance</li> </ul> <p>This problem prevented interoperability with other ORBs.</p>

# RMI

---

<b>Change Request Number</b>	<b>Description</b>
CR096511	<p>Under WebLogic Server 6.1 SP03 and SP04, out of memory messages occurred on repeated creation and close of InitialContext.</p> <p>This problem was diagnosed as a memory leak on the <code>DGCServerHelper</code> class.</p> <p>This problem was resolved with a code change to register <code>LocalServerRef</code> only if the <code>dgcpolicy</code> on descriptor is not “Managed”.</p>
CR089481	<p>Under WebLogic Server 6.1 SP03, a client JDBC connection that was not closed properly by the client, was not closed by WebLogic Server and garbage collected.</p> <p>This problem was solved by a code fix to ensure that connections are be closed properly even when the remote client does not close the connection properly.</p> <p><b>Notes:</b> Client-side code should properly close connections. Although WebLogic Server will close unclosed connections, the process may be protracted, depending on the use and the distributed garbage collection behavior.</p>
CR100430	<p>In WebLogic Server 6.1 SP04, with a WebLogic Server 8.1 server instance as client of a WebLogic Server 6.1 SP04 server instance, a transaction timed out with this exception:</p> <pre>java.rmi.RemoteException: Exception while committing Tx: Name=[EJB weblogic.qa.tests.interop.ejb11.txcrossdomain.TravelPlannerBean.planItinerary_Required(java.lang.String,java.util.Properties)],Xid=BEA1-000147799CCFA039D1D7(2653792),Status=Committing,numRepliesOwedMe=0,numRepliesOwedOthers=0,seconds since begin=120,seconds</pre> <p>This interoperability issues was solved with a code fix.</p>

---

# Security

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

CR090101 WebLogic Server did not ensure each certificate in a certificate chain was issued by a certificate authority. This problem meant anyone could get a personal certificate from a trusted certificate authority, use that certificate to issue other certificates, and WebLogic Server would not detect the invalid certificates. Now all X509 V3 CA certificates used with WebLogic Server must have the Basic Constraint extension defined as CA thus ensuring all certificates in a certificate chain were issued by a certificate authority. By default, any certificates for certificate authorities not meeting this criteria are rejected. If WebLogic Server is booted with a certificate chain that will not pass the certificate validation, an information message is logged noting that clients could reject it. For more information, see [SSL Certificate Validation](#).

---

CR051018 In previous releases of WebLogic Server, the system password was stored in clear text. The password is now encrypted and stored in `password.ini`.

**Note:** `password.ini` cannot be upgraded to the `boot.properties` file used in WebLogic Server version 7.0.

---

CR100592 WebLogic Server 6.1 SP04 became unusable after IPlanet LDAP realm disabled a user. The problem appeared with this sequence of events:

Bring up an LDAP realm with the server. Deploy a webapp that uses form authentication. For any user in the LDAP realm, use the wrong password until you reach the user lockout threshold. Client receives a 500 error, and the server will show:

```
<Mar 9, 2003 7:17:22 PM PST> <Error> <HTTP>
<[WebAppServletContext(3017390,security,/security)] Servlet
failed with Exception netscape.ldap.LDAPException: error
result (19); Exceed password retry limit. Please try later.;
Constraint violation
```

After this error, the server becomes unusable with respect to authentication; for example, no other user can log into the web application, and the console is inaccessible.

The problem was solved with a code fix to check for LDAP exceptions when user enters an invalid password.

---

## Servlets

Change Request Number	Description
CR077922	<p>During testing, the session binding listener's (<code>HttpSessionBindingListener</code>) <code>valueBound()</code> and <code>valueUnbound()</code> methods were called multiple times for a single invocation.</p> <p>A code fix was implemented to prevent duplicate method calls.</p>
CR083624	<p>A new configuration parameter, <code>UseHighestCompatibleHTTPVersion</code>, was added to the <code>WebServer</code> element in <code>config.xml</code>. This parameter causes WebLogic Server to send responses in the highest version of the HTTP protocol that is compatible with the version used in the request. For instance, setting this parameter to be true causes WebLogic Server to respond to an HTTP/1.0 request with an HTTP/1.1 response. This affects only the version-string portion of the response. The default setting for the parameter is false in WebLogic Server 6.1</p> <p>The parameter value can be changed by editing <code>config.xml</code>. The Administration Console does not provide an interface for setting the parameter value.</p>
CR086579	<p>Using the NSAPI proxy plug-in with form-based authentication through iPlanet listening on port 80, <code>j_security_check</code> was not redirecting correctly, and authentication failed.</p> <p>Analysis revealed that the form based authentication module did not call <code>processRedirectedURL</code> in the case of default port.</p>
CR086587	<p>In WebLogic Server 6.1 SP03, the web server log (<code>access.log</code>), and the JDBC sub-system log (<code>jdbc.log</code>) were created relative to the product installation directory, instead to the <code>RootDirectory</code>.</p> <p>This problem was corrected with a code fix.</p>
CR087573	<p>In WebLogic Server 6.1 SP01, when using JDBC persistence for a session and storing a complex object that contained nested <code>hashMaps</code> and <code>TreeMaps</code>, a subsequent request came before the session was persisted, resulting in loss of session data.</p> <p>The problem was solved with a code fix to move <code>removeSession(id)</code> to the end of method <code>sync()</code> in <code>FileSessionContext.java</code>.</p>
CR087647	<p>The fix associated with “<a href="#">CR055987</a>” introduced a timeout to solve the connection not being released problem. The fix introduced an <code>InterruptedException</code> that was not handled. When the exception was encountered, unwanted retries occurred.</p> <p>The problem was resolved with a code fix to catch <code>InterruptedException</code> and re-throw the exception to prevent retries.</p>

CR087984 In WebLogic Server 6.1SP02, `ServletContextListener` was invoked before `HttpSessionListener`. According to the 2.3 servlet specifications, session listeners must be notified of session invalidations prior to context listeners being notified of application shutdown.

A code fix was made to ensure compliance with the specification.

---

CR088166 In WebLogic Server 6.1SP03, a web application generated a `java.lang.ClassNotFoundException` without indicated the name if the missing file:

```
<[WebAppServletContext(4730538,olam,/olam)] Error loading
servlet: "> java.lang.ClassNotFoundException:
```

Analysis revealed that WebLogic Server did not correctly handle a null or empty string value for the `auth-filter` tag. The problem was resolved with a code fix.

---

CR088350 Using `HttpProxyServlet` to proxy to a 3rd party event server from KnowNow caused Internet Explorer 6 to wait indefinitely with no response.

Analysis revealed that the KnowNow server :

- only responds to HTTP 1.0 requests
- does not send the `ContentLength`

If there is no content length, and the data is not chunk transferred, WebLogic Server reads until the end of stream. If the KnowNow server keeps sending data, `HttpProxyServlet` keeps reading data.

The problem was resolved with a new configuration parameter, `UseKnowNow`, that causes `HttpProxyServlet` to:

- read and write byte by byte, and
- use HTTP/1.0 to send requests to KnowNow

To use this parameter, add this syntax to the `web.xml` for the `HttpClusterServlet` `webapp`.

```
<init-param> <param-name>UseKnowNow</param-name>
<param-value>true</param-value> </init-param>
```

---

CR088384 In WebLogic Server 6.1SP02, `getContextPath` from `HttpServletRequest` returned incorrect value. For instance, for the request

```
http://localhost/webapp/foo
getContextPath returned
/webap
```

This error was corrected with a code fix.

---

---

CR088478	<p>The <code>MaxSkips</code> parameter for <code>HttpClusterServlet</code> and all plug-ins has been deprecated, and replaced with the <code>MaxSkipTime</code> parameter.</p> <p>With this parameter,</p> <ul style="list-style-type: none"><li>■ when the plug-in marks a server “bad”, it notes the time</li><li>■ the bad server is skipped in load balancing cycles until <code>MaxSkipTime</code> is reached</li><li>■ then, a new server list is forced</li></ul>
CR088747	<p>Session was lost when trying to re-establish a WebLogic Server session after a request to a web application that did not explicitly set <code>CookiePath</code> in <code>weblogic.xml</code>.</p> <p>Analysis revealed the browser (IE6) sent back two session cookies with different paths:</p> <ul style="list-style-type: none"><li>■ <code>JSESSIONID</code> with <code>/webapp</code> as the path</li><li>■ <code>JSESSIONID</code> with <code>/</code> as the path</li></ul> <p>According to J2EE, multiple cookies in the header are ordered such that those with more specific <code>Path</code> attributes precede those with less specific. WebLogic Server searched the cookie vector backwards for the <code>JSESSIONID</code>, for performance reasons, and hence encountered the less specific path .</p> <p>A code change was implemented to ensure that WebLogic Server uses the first cookie in the header.</p>
CR088759	<p>In WebLogic Server 6.1 SP03, <code>redirect-with-absolute-url = false</code> did not work for <code>j_security_check</code>.</p> <p>This is because <code>j_security_check</code> stores an absolute path and passes it to the <code>sendRedirect()</code> method.</p> <p><code>j_security_check</code> also should be aware of <code>redirect-with-absolute-url</code>.</p> <p>A code change was implemented to make <code>FormSecurityModule</code> redirect requests in accordance with the setting of <code>redirect-with-absolute-url</code>.</p>
CR089582	<p>In WebLogic Server 6.1 SP03 under Solaris 8, when receiving a HTTP/1.1 request that did not have a <code>Host</code> header field, the value of <code>Date</code> header field that WebLogic Server returned was "Date: Thu, 01 Jan 1970 00:00:00 GMT". This made it impossible to determine when the erroneous request was generated.</p> <p>The problem was solved by a code change to set the request invoke time before sending the bad request error.</p>
CR089868	<p>A performance fix introduced in WebLogic Server 6.1 Service Pack 3 (“083654” on page 6-246) caused multi-byte headers to become garbled. A work-around is available, but in SP03 and SP04, a patch was required to employ the work-around. (See “089868” on page 5-33 for details.)</p> <p>In SP05, the CR089868 patch is no longer necessary.</p>

---

CR090033 WebLogic Server 6.1 SP03 threw a null pointer exception while parsing request when the WL-Proxy-SSL header precedes the Host header. This occurred with client running over HTTPS, and SSL accelerator, IPlanet, and WebLogic Server running over HTTP. The problem was only replicated when configuration included the SSL accelerator.

```
java.lang.NullPointerException at
weblogic.servlet.internal.ServletRequestImpl.setField(ServletRequestImpl.java:1903) at
weblogic.servlet.internal.RequestParser.parse(RequestParser.java:254) at
weblogic.servlet.internal.MuxableSocketHTTP.dispatch(MuxableSocketHTTP.java:390) at
weblogic.socket.MuxableSocketDiscriminator.dispatch(MuxableSocketDiscriminator.java:275) at
weblogic.socket.NTSSocketMuxer.processSockets(NTSSocketMuxer.java:633) at
weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:23) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:152) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:133)
```

The error occurred because the context is null, as the request was still being parsed, and context was not yet known. The WL-Proxy-SSL went before the HOST header. Problem was resolved with a code fix.

---

CR090188 For chunked data transfer WebLogic Server 6.1 SP03 prepended the chunk size to the data to be transferred, hence adding extra characters to the data transferred.

The problem was resolved with a code change, to move `setChunking` out of `mergePostParams` of `ServletRequestImpl` to `MuxableSocketHttp` after `request.setInputStream`.

With this change, servlets and JSPs no longer need to decode chunk data. The WebLogic Server servlet engine will merge chunk data automatically.

---

CR090225 Please review the security advisory information at <http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-14.jsp>.

---

CR090465 When hosting a web application with user-defined error pages, WebLogic Server returned the following response for 400 error and closed the connection immediately, with no `Connection: close` header:

```
HTTP/1.1 400 Bad Request Date: Thu, 01 Jan 1970 00:00:00 GMT
Content-Length: 463 Content-Type: text/html Last-Modified:
Thu, 07 Nov 2002 21:56:52 GMT <!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 3.2//EN"> <html>
```

The problem was solved with a code fix.

---

- CR090555      WebLogic Server threw a null pointer exception to `HttpClusterServlet` when checking for `X_WEBLOGIC_CLUSTER_HASH`. Stack trace:
- ```
<Tue Nov 05 15:58:00 PST 2002>: Start connection timeout scheduler <Tue Nov 05 15:58:00 PST 2002>: GenericProxyServlet: init() <Tue Nov 05 15:58:00 PST 2002>:
HttpClusterServlet:init() <Tue Nov 05 15:58:00 PST 2002>:
===New Request===GET /base;JSESSIONID=9G8ZGUTNwnNTXm8B5V5z41ehG0T3oChTPvhnXe7EHCZ1IQI5lgF4!822557425 HTTP/1.1 <Tue Nov 05 15:58:00 PST 2002>: Found cookie: 822557425 <Tue Nov 05 15:58:00 PST 2002>: Trying to get JVMID id from server: coolant!7003!443 <Tue Nov 05 15:58:00 PST 2002>: Update dynamic hash: WakHYk8LlxQ7XrnhkSXEohfZMAw <Tue Nov 05 15:58:00 PST 2002>: java.lang.NullPointerException at
weblogic.servlet.proxy.HttpClusterServlet.initDynamicServerList(HttpClusterServlet .java:761) at
weblogic.servlet.proxy.HttpClusterServlet.getPreferred(HttpClusterServlet.java:622) at
weblogic.servlet.proxy.HttpClusterServlet.service(HttpClusterServlet.java:21
```
- Analysis revealed that response headers are stored in the `HashMap` in random order. A code fix was implemented to set the server list before setting hash.
- 
- CR090665      WebLogic Server 6.1 SP03, servlet filters had to explicitly call `flushBuffer()` on the response wrappers after having called `chain.doFilter()` to send the response to the client. This caused customers to have to write separate versions of the filter for different application servers. Not doing so causes the response to contain no data. This happens because of bytes being buffered in the `printWriter's OutputStreamWriter` object.
- A code change was implemented to ensure that user code does not need to explicitly call `flushBuffer()`.
- 
- CR091410      When "extended format" is selected for the HTTP logging option and the server is restarted, the following Null Pointer Exception was thrown:
- ```
<Oct 9, 2002 6:45:48 PM PDT> <Error> <kernel> <000802>
<ExecuteRequest failed
  java.lang.NullPointerException
  java.lang.NullPointerException
```
- The problem was solved with a code fix to open `access.log` file if `filename` is null
-

CR091759      `RequestDispatcher.forward` dropped the version number at the end of the URL header. The request should have been forwarded as:  
Request URI: `/RequestDispatcher/SnoopServlet/myTest.txt;37`  
Instead, it was forwarded as:  
Request URI: `/RequestDispatcher/ProxySnoopServlet/myTest.txt`  
The problem was resolved to a code fix to set `processPathParameters` to false in `setRequestURI()` when doing `forward()`.

---

CR091878      When `HttpClusterServlet` was undeployed during a load test, an `.IndexOutOfBoundsException` resulted:  
<Nov 27, 2002 4:05:11 PM PST> <Error> <HTTP> <101268>  
<ServletContext(id=5220998,name=proxywebapp,context-path=):  
Failed while destroying servlet:  
java.lang.IndexOutOfBoundsException: Index: 2, Size: 2  
IndexOutOfBoundsException while destroying HttpClusterServlet  
Analysis revealed an error occurred when removing the last element in the connection pool with `pool.remove(i)`.  
The problem was resolved with a code fix.

---

CR092255      WebLogic Server 6.1 SP04 ignored the quote character (") if cookie valued includes a quote character. For example, this syntax:  
`Cookie cookieWithQuotedValue = new Cookie("TestQuotedCookie",  
"\TestValue\");`  
assigned `TestValue` is added as cookie value, not `"TestValue"`.  
A code change was made to preserve quote characters in cookie values.

---

CR092428      In WebLogic Server 6.1 SP04, comma character (,) in a Netscape cookie resulted in a `Got bad cookie header` error message.  
A code fix resolved the problem.

---

CR092778      In WebLogic Server 6.1 SP04, JISAutoDetect encoding for HttpRequest resulted in this `UnsupportedEncodingException`:

```
java.io.UnsupportedEncodingException: JISAutoDetect at
sun.io.Converters.getConverterClass(Converters.java:102) at
sun.io.Converters.newConverter(Converters.java:133) at
sun.io.CharToByteConverter.getConverter(CharToByteConverter.
java:62) at
weblogic.servlet.internal.ServletRequestImpl.setCharacterEnc
oding(ServletRequestImpl.java:344) ...
```

This problem did not occur with WebLogic Server 6.1 SP03.

The problem occurred because `ServletRequestImpl.java` used the `CharToByteConverter` class instead of `sun.io.ByteToCharConverter`. `CharToByteConverter` is the one for input converter. `CharToByte` is for output converter. `JISAutoDetect` is only available for input stream.

The problem was corrected with a code fix.

---

CR093167      In WebLogic Server 6.1 SP03, when using the `HTTPClusterServlet` to access a secure webapp on a backend cluster, Internet Explorer 5.5 hung for 30 seconds after the username and password were entered. The problem was not replicated in other browsers, and did not occur under Internet Explorer when accessing a back end server directly, rather than through the plug-in.

The problem occurred because Internet Explorer does not close the connection when it gets a `Connection: Close` header from the server. When accessing WebLogic Server directly, WebLogic Server closes the connection when it throws a 401 error. However through the `HttpProxyServlet`, the browser-to-ProxyServlet connection was kept alive even though the backend server had closed the connection.

The problem was solved by a code fix to `GenericProxyServlet` which causes the front end connection to disable `KeepAlives` when a `Connection: Close` header is returned from the back end server. This causes Internet Explorer to send the authentication details back straight away instead of hanging while waiting for WebLogic Server to timeout the connection.

---

CR093209

A new installation of WebLogic Server 6.1 SP04 threw a null pointer exception when a successfully deployed web app that has a property set to java protocol is accessed. The browser returned Error 500--Internal Server Error. The problem did not occur with an upgrade installation.

```
<Dec 16, 2002 11:59:05 AM PST> <Error> <HTTP>
<[WebAppServletContext(2092664,sampleApp,/sampleApp)]
Servlet failed with
Exception
java.lang.NullPointerException
    at
weblogic.servlet.JSPServlet.service(JSPServlet.java:132)
    at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
    at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:262)
    at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Serv
letStubImpl.java:198)
    at
weblogic.servlet.internal.WebAppServletContext.invokeServlet
(WebAppServletContext.java:2637)
    at
weblogic.servlet.internal.ServletRequestImpl.execute(Servlet
RequestImpl.java:2359)
    at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
)
    at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
>
```

Analysis revealed an error in `ZipSource.getURL()`. The problem was resolved by adding logic to check for null value of `source.getURL()`.

---

## 6 Resolved Problems

---

---

CR093634	<p>In WebLogic Server 6.1 SP03 the <code>HttpRequest.getRemoteAddr()</code> method returned the IP address of <code>HttpClusterServlet</code> instead of the client browser's IP address.</p> <p>Analysis revealed that <code>HttpClusterServlet</code> does not set the proprietary header, "WL-Proxy-Client-IP", and instead returns the IP address of the socket upon which it is listening.</p> <p>The problem was solved by a code fix to <code>HttpClusterServlet</code> to set the "WL-Proxy-Client-IP" header when <code>WebLogic Plugin Enabled</code> parameter is enabled.</p>
CR093755	<p>The error message:</p> <pre>&lt;Warning&gt; &lt;HTTP&gt; &lt;WebAppServletContext(5294080,blue,/blue) One of the getParameter family of methods called after reading from the ServletInputStream, not merging post parameters&gt; was erroneously generated.</pre> <p>A code fix was implemented to ensure that the message is not generated erroneously.</p>
CR094488	<p>A new capability was added to allow a user to securely access HTTPS resources in a session that was initiated using HTTP, without loss of session data. To enable this new feature, add <code>AuthCookieEnabled="true"</code> to the <code>WebServer</code> element in <code>config.xml</code>:</p> <pre>&lt;WebServer Name="myserver" AuthCookieEnabled="true"/&gt;</pre> <p>This will cause a new secure cookie to be sent to the browser when authenticating via an HTTPS connection. Once set, the session can access other security-constrained HTTPS resources only if the cookie is sent from the browser.</p> <p><b>Note:</b> If authenticating via plain HTTP, the secure cookie will not be set or required for any HTTPS resources. When accessing a non-protected HTTPS resource, the cookie will not be verified (since it will not have been sent from the browser). This allows the browser to access non-protected HTTPS resources without the user logging in.</p>
CR094773	<p>Duplicate of "<a href="#">CR094561</a>".</p> <p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourceLibrary/advisoriesnotifications/BEA03-26.01.jsp">http://dev2dev.bea.com/resourceLibrary/advisoriesnotifications/BEA03-26.01.jsp</a>.</p>

---

CR094997      In WebLogic Server SP02, when two interacting web applications use different session cookie names, the session ID was lost.

Analysis revealed that session attributes were not updated correctly when a web application specifies `cookieName`, a resource from another web application is included, and a `RemoteUser` exists in the request prior to the inclusion of the external resource.

The problem was solved by a code fix.

---

CR095548      During testing, session data was lost intermittently, when the persistence type was JDBC. The following exception occurred:

```
weblogic.qa.tests.cluster.webapp.plugins.RequestSessionBefore
eInvalidateTest.testRequestSessionBeforeInvalidationTest()
Session Was Invalidated
Sent:
http://qa47-1:7771/pluginsjdbc/SessionBeforeInvalidate.jsp?c
ount=1
Received: <html><body>PROBLEM: session is not the same as the
first
request!FirstRequest sid:
2tjgF02nPIGgHGyOjFubyOIjqt0ZAVK4nolNy0afTxGKMroJ25z5!2356550
501043194848402Sid
```

The problem was resolved with a code fix to the `isRequestedSessionIdValid()` method.

---

CR095570

WebLogic Server 6.1 SP04 threw a `NullPointerException` when a null value was set in a servlet HTTP header:

```
<Jan 22, 2003 11:11:53 AM KST> <Error> <Kernel> <Execute
Request failed java.lang.NullPointerException: Start server
side stack trace: java.lang.NullPointerException at
weblogic.servlet.internal.ResponseHeaders.writeHeaders(Respo
nseHeaders.java:360) at weblogic.servlet.internal.
ServletResponseImpl.writeHeaders(ServletResponseImpl.java:90
2) at weblogic.servlet.internal.ServletOutputStreamImpl.
sendHeaders(ServletOutputStreamImpl.java:239) at weblogic.
servlet.internal.ServletOutputStreamImpl.flush(ServletOutput
StreamImpl.java:121) at weblogic.servlet.internal.
ServletOutputStreamImpl.commit(ServletOutputStreamImpl.java:
484) at weblogic.servlet.internal.ServletOutputStream
Impl.finish(ServletOutputStreamImpl.java:531) at weblogic.
servlet.internal.ServletResponseImpl.send(ServletResponseImp
l.java:1091) at weblogic.servlet.internal.Servlet
RequestImpl.execute(ServletRequestImpl.java:2364) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139
) at weblogic.kernel.ExecuteThread.run(ExecuteThread.
java:120) End server side stack trace at weblogic.servlet.
internal.ResponseHeaders.writeHeaders(ResponseHeaders.java:3
60) at weblogic.servlet.internal.ServletResponseImpl.
writeHeaders(ServletResponseImpl.java:902) at weblogic.
servlet.internal.ServletOutputStreamImpl.sendHeaders(Servlet
OutputStreamImpl.java:239) at weblogic.servlet.internal.
ServletOutputStreamImpl.flush(ServletOutputStreamImpl.java:1
21) at weblogic.servlet.internal.ServletOutputStream
Impl.commit(ServletOutputStreamImpl.java:484) at
weblogic.servlet.internal.ServletOutputStreamImpl.finish(Ser
vletOutputStreamImpl.java:531) at weblogic.servlet.internal.
ServletResponseImpl.send(ServletResponseImpl.java:1091) at
weblogic.servlet.internal.ServletRequestImpl.execute(Servlet
RequestImpl.java:2364) at weblogic.kernel.ExecuteThread.
execute(ExecuteThread.java:139) at weblogic.kernel.Execute
Thread.run(ExecuteThread.java:120)
```

The problem was resolved by added logic to check for null values in `ResponseHeaders.java`.

---

CR096459      Response time for a web application deployed to WebLogic Server 6.1 SP02, when accessed from a browser set to HTTP Version 1.0 with `flushRes=true`. The problem did not occur with `flushRes=false`.

Analysis revealed that `useKeepAlive` was reinitialized inappropriately. The socket connection was not closed when it should have been, and the client waited for the socket to time out.

A code fix was implemented to initialize `useKeepAlive` when setting request and response.

---

CR097719      In WebLogic Server SP02, SP03, and SP04, when getting certain POST requests from WAP devices, a web application encountered a

```
java.util.ConcurrentModificationException
<29.jan.03 13:03:05 WET> <Error> <HTTP>
<[WebAppServletContext(2810713,TnMFF,/TnMFF)] Servlet failed
with Exception java.util.ConcurrentModificationException at
java.util.HashMap$HashIterator.next(HashMap.java:736) at
weblogic.utils.enumerations.IteratorEnumerator.nextElement(IteratorEnumerator.java:25) at
com.colibria.core.xmlswitch.XMLSwitch.getQueryStringXML(XMLSwitch.java:347) at ...
```

Analysis revealed that when a charset is associated with the request's `content-type`, the code read the data again using the encoding, and reset the query parameters, in the application. The application already has the enumeration object and iterates it (`request.getParameterNames`), and then tries to get the value of the parameter (`request.getParameterValues(k)`). At that point, the query parameters are wiped out and `getParameterValues` method tries to set it, resulting in the exception.

The problem was resolved by a code change to set query parameters after they are wiped, so that the data can be read again when a charset is associated with the request's `content-type`.

---

CR098955      The scheduled log rotation time for a WebLogic Server HTTP server log was calculated incorrectly when `weblogic.httpd.logRotationBeginTime` was a date in the past.

The error was corrected by changing the data type of operands used in the calculation from `INT` to `LONG`.

---

CR099984      The input stream was consumed by the servlet engine before the servlet code could access it. This problem was resolved by a code change to ensure that internal reads do not consume the input stream.

---

CR100265      If the request received has no cookie but does have valid credentials WebLogic Server should allow access to the protected resource. Instead, this exception occurred:

```
java.lang.NullPointerException at
weblogic.servlet.security.internal.SecurityModule.checkAuthC
ookie(SecurityModule.java:579) at
weblogic.servlet.security.internal.BasicSecurityModule.check
UserPerm(BasicSecurityModule.java:157) at ...
```

The problem was solved with a code fix.

---

CR100572      When a request with an incorrect URI was received from a from plug-in, WebLogic Server threw this stack trace

```
<Mar 8, 2003 3:27:20 PM PST> <Error> <Socket> <BEA-000421>
<Uncaught Throwable in processSockets java.lang.NullPointer
Exception. java.lang.NullPointerException at weblogic.
servlet.internal.ServletResponseImpl.writeHeaders(ServletRes
ponseImpl.java:968) at weblogic.servlet.internal.Servlet
OutputStreamImpl.sendHeaders(ServletOutputStreamImpl.java:24
4) at weblogic.servlet.internal.ServletOutputStreamImpl.
flush(ServletOutputStreamImpl.java:121) at weblogic.servlet.
internal.ServletOutputStreamImpl.commit(ServletOutputStreamI
mpl.java:486) at weblogic.servlet.internal.ChunkOutput.
commit(ChunkOutput.java:269) at weblogic.servlet.internal.
ChunkOutputWrapper.write(ChunkOutputWrapper.java:91) at
weblogic.servlet.internal.ChunkWriter.write(ChunkWriter.java
:37) at java.io.Writer.write(Writer.java:150) at
java.io.PrintWriter.write(PrintWriter.java:230) at
java.io.PrintWriter.write(PrintWriter.java:247) at
java.io.PrintWriter.print(PrintWriter.java:378) at
weblogic.servlet.internal.ServletResponseImpl.sendError(Serv
letResponseImpl.java:420) at weblogic.servlet.internal.
ServletResponseImpl.sendError(ServletResponseImpl.java:373)
at weblogic.servlet.internal.MuxableSocketHTTP.dispatch
(MuxableSocketHTTP.java:512) at weblogic.socket.MuxableSocket
Discriminator.dispatch(MuxableSocketDiscriminator.java:281)
at weblogic.socket.NTSocketMuxer.processSockets(NTSocket
Muxer.java:102) at weblogic.socket.SocketReaderRequest.
execute(SocketReaderRequest.java:32) at weblogic.kernel.
ExecuteThread.execute(ExecuteThread.java:178) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:151)
```

This problem was solved with a code fix.

---

CR100172

When multiple chunked requests are posted to a servlet a number of the requests fail with a `NumberFormatException`. Mostly of the time alternate requests are fulfilled and the next request fails. It seems to be occurring as the requests are being made on the same connection. The exception stack trace is:

```
<ExecuteThread: '11' for queue: 'default'> <kernel identity>
<> <101017>
<[ServletContext(id=4864139,name=387551,context-path=)] Root
cause of ServletException> java.lang.NumberFormatException:
at java.lang.Integer.parseInt(Integer.java:430) at
weblogic.utils.http.HttpChunkInputStream.readChunkSize(HttpC
hunkInputStream.java:118) at
weblogic.utils.http.HttpChunkInputStream.initChunk(HttpChunk
InputStream.java:72)
```

Analysis revealed two problems in `PostInputStream`:

- did not reset control variable `readAllChunks` when recycling `PostInputStream`,
- `isChunkComplete` did not detect the end of chunk correctly

The problem was solved by a code fix to read until end of stream in `PostInputStream` and correctly detect end of chunk in `HttpChunkInputStream`.

---

CR100837

WebLogic Server 6.1 SP04 misdirected output for JSP includes in JSP pages.

The problem was corrected with a code fix.

---

---

## WLS-Tour/Examples

---

<b>Change Request Number</b>	<b>Description</b>
CR087117	<p>In WebLogic Server 6.1 SP04, a password error occurred when starting the Administration Server, the Petstore server and the Example server, before the user entered a password:</p> <pre>Starting WebLogic Server .... &lt;Oct 1, 2002 9:22:10 AM EDT&gt; &lt;Notice&gt; &lt;Management&gt; &lt;Loading configuration file .\config\petstore\config.xml ...&gt; &lt;Oct 1, 2002 9:22:13 AM EDT&gt; &lt;Emergency&gt; &lt;Security&gt; &lt;Authentication failure - reenter password to boot weblogic server:&gt;</pre> <p>A code fix solved this problem.</p>
CR087366	<p>In WebLogic Server 6.1 SP04, the "WebLogic Server Code Examples Index" link in the samples server index page (<code>index.jsp</code>) did not work. When a the user clicked the link, the following message was displayed:</p> <pre>"Netscape is unable to find the file or directory named /D:/bea61. Check filename and try again.</pre> <p>Analysis revealed that WebLogic Server had been installed with a BEAHOME that contained a blank space.</p> <p>The problem was resolved by adding double quotes to the relevant href in <code>index.jsp</code> to allow for space characters in BEAHOME.</p>
CR090776	<p>An error occurred when deploying Petstore applications during an upgrade from WebLogic Server 6.1 SP04 to WebLogic Server 8.1.</p> <p>Analysis revealed that deployment descriptors lacked EJB references. The missing references were caught by the WebLogic Server 8.1, but not by WebLogic Server 6.1</p> <p>This problem was resolved by corrections to the Petstore deployment descriptors.</p>
CR092881	<p>In WebLogic Server 6.1 SP04 on Unix AIX, When a user tried to start petstore with <code>StartPetStore.sh</code> this error resulted:</p> <pre>java.lang.SecurityException: Unable to locate a login configuration.</pre> <p>The problem was solved with a correction to <code>StartPetStore.sh</code>.</p>

---

- 
- CR096147      `dmloadcf` failed for `dom1config` for WebLogic Server Samples `simpFML32`  
Analysis revealed that `*DM_LOCAL_SERVICES` contained:  
`*DM_LOCAL_SERVICES`  
`REVERSE_STRING RDOM="TDOM1"`  
instead of:  
`*DM_LOCAL_SERVICES`  
`REVERSE_STRING LDOM="TDOM1"`  
An error in the Tuxedo `dom1config` configuration file was corrected to allow Tuxedo to export the local service, `REVERSE_STRING`, provided by local domain `TDOM1`.
- 
- CR102138      The documentation for the SSL Client example were incorrect. The instructions directed the user to copy the `WL_HOME\lib\cacerts` to `JAVA_HOME\jre\lib\security\jssecacerts`.  
This problem was solved by correcting the instructions, adding the `jssecacerts` file to the `sslclient` sample directory, and adding sample output to the instructions for the JSSE sample.
-

## Web Services

---

<b>Change Request Number</b>	<b>Description</b>
CR087529	<p>In WebLogic Server 6.1SP03, the client for the sample web service at <code>samples\examples\webservices\rpc\weatherEJB</code> was built by <code>wsgen</code> and deployed using only the <code>client.jar</code> (no <code>weblogic.jar</code>). When the non-SSL web service was invoked by the Java client, the following exception was thrown:</p> <pre>&lt;&lt; Missing class files in servicegen client.jar. Getting Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/net/http/URLConnection at weblogic.soap.http.SoopContext.lookup(SoopContext.java:87) at javax.naming.InitialContext.lookup(InitialContext.java:350) at com.verizon.client.client.main(client.java:72) &gt;&gt;</pre> <p>Analysis revealed that the <code>weblogic.net.http.URLConnection</code> class is necessary for minimal out-of-the-box functionality and should be included in the <code>client.jar</code>.</p> <p>The problem was solved by adding <code>weblogic.net.http.URLConnection</code> to <code>client.txt</code></p>
CR091862	<p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resource/library/advisories/notifications/BEA02-23.jsp">http://dev2dev.bea.com/resource/library/advisories/notifications/BEA02-23.jsp</a>.</p>

---

CR095044

In WebLogic 6.1 SP03, when `org.apache.soap.Fault` was used (to represent the contents and semantics of a `<SOAP-ENV:Fault>` element), the `Fault.getDetailEntries()` method call should result in a Vector of exceptions, when the web service operation throws an exception.

Instead, in WebLogic 6.1 SP03, this call returns a null and the client does not get to see the exception stack trace. This problem does not exist in SP02.

The client runs with the same CLASSPATH while hitting either the WebLogic 6.1 SP02 or WebLogic 6.1 SP03 server instances. The problem was reproduced with the client CLASSPATH:

```
CLASSPATH=.;.\soap.jar;.\xerces.jar;.\j2ee.jar
```

The difference in behavior is that in SP03, the content of the fault detail is wrapped in `<![CDATA[ ]>`. This is done to prevent the parser from failing if the stack trace contains something like

```
<<no stacktrace available>>
```

The problem was resolved by the addition of a new servlet initialization parameter, `cdata-fault-detail`, which if set to false, causes fault detail to not be wrapped in `<![CDATA[ ]>`

```
<servlet> ...
```

```
<servlet-class>weblogic.soap.server.servlet.StatelessBeanAdapter</servlet-class> <init-param>
<param-name>cdata-fault-detail</param-name>
<param-value>>true</param-value> </init-param> ...
```

If `cdata-fault-detail` to false, if the stack trace contains something like

```
<<no stacktrace available>>
```

a parser error will result.

CR096906

WebLogic 6.1 SP04, did not return `dateTime` entries correctly when a web service generated by WebLogic Server was called. The "+" char was omitted from the time zone offset part of the `dateTime` entry, when the time zone offset is positive (i.e., GMT+1 = CET [time zone for Denmark]). Example:

February 3rd, 2003 09:12:04 CET should be `dateTime`:

```
<date xsi:type="xsd:dateTime">2003-02-03T09:12:04.000+01:00
</date>
```

Weblogic sent it as:

```
<date xsi:type="xsd:dateTime">2003-02-03T09:12:04.00001:00
</date>
```

The problem was resolved with a code fix.

---

## 6 *Resolved Problems*

---

---

CR098364      In WebLogic 6.1 SP03, when Ant was used to generate a .war file with Web Services using this WebLogic task:

```
<taskdef name="wsgen"
classname="weblogic.ant.taskdefs.ejb.WSGen" />
```

and the `Transactions.jar` file had more than 155 EJBs, the `wsgen` failed.

The problem was resolved by a code fix to `CompilerInvoker`.

---

CR099255      In WebLogic 6.1 SP04, in the `weblogic.soap.WebServiceProxy` class, the `invoke()` method did not check if certs had been set in the class.

This caused two-way authentication to fail, or alternatively if the certs are passed to the `InitialContext` the http protocol did not work.

The problem was solved with a code fix.

---

# WebLogic Tuxedo

Change Request Number	Description
CR084435	<p>In WebLogic 6.1 SP02 WTC connection policies did not work properly in failover and failback situations when the connection policy was set to ON_DEMAND and ON_STARTUP. When the connection policy was set to ON_STARTUP, failover doesn't work. Attempts to reproduce the problem revealed related errors:</p> <ol style="list-style-type: none"> <li>1. The WTC service request failed over to the secondary domain listed in the services section for the <code>bdmconfig</code> file.</li> <li>2. The WTC service request did not check the primary domain listed in the services section of the <code>bdmconfig</code> file; the request went to the secondary domain.</li> </ol> <p>When the primary node became unavailable as configured in <code>bdmconfig.xml</code>, the service request failed over to the secondary server in the imported services section:</p> <pre>&lt;T_DM_IMPORT ResourceName="TOUPPER" LocalAccessPoint="WLSDOM" RemoteAccessPointList="SIMPDOM,TESTDOM"&gt; &lt;TranTime&gt;600&lt;/TranTime&gt; &lt; /T_DM_IMPORT&gt;</pre> <p>When SIMPDOM became unavailable, the service request was fulfilled by TESTDOM—this should not happen when the connection policy is ON_DEMAND.</p> <p>Analysis indicated that the connection policy wasn't being checked. The problem was resolved with a code fix to <code>weblogic/wtc/gwt/TuxedoConnection.java</code>.</p>
CR095588	<p>In the <code>simpcnv</code> example, the Tuxedo client failed to contact WebLogic Server. The client hit the local Tuxedo server instead of WebLogic Server.</p> <p>Code changes were made to the Tuxedo client so that it does not look for padding. In addition, a remote services section was added to <code>dubb</code>, and an export section was added to <code>bmdconfig.xml</code> file.</p>
CR092860	<p>Using WTC to access Tuxedo 8 Corba objects resulted in a <code>JavaIDL Reader</code> thread leak. Running a simple example application resulted in a build-up of <code>JavaIDL Reader</code> threads in the JVM running WebLogic Server. The threads were not destroyed until the server was shut down.</p> <p>Analysis revealed the <code>finalize()</code> was not called at the correct time. A code fix to add a call to <code>ORB.destroy()</code> was implemented to resolve the problem.</p>

---

CR090137	In WebLogic Server 6.1 SP02, the Xalan engine in WebLogic Server produced an invalid XML file, containing an unmatched <META> tag. A code fix to apache/xalan code solved the problem.
CR074963	When creating large FML tables (on the order of 4K entries), <code>mkfldclass[ 32 ]</code> generated a method to instantiate each entry into a hash table. For large FML tables, this single method exceeds the size limitation in the JVM. This problem was resolved by implementing a dynamic load field table mechanism.

---

## XML

---

Change Request Number	Description
CR091862	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-23.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-23.jsp</a> .

---

# WebLogic Server 6.1 Service Pack 4 Solutions

The following sections describe problems that have been resolved with WebLogic Server 6.1 Service Pack 4.

- Classloader
- Cluster
- Console
- EJB
- Examples
- JDBC and jDriver
- JMS

- Miscellaneous
- Plug-ins
- Servlets, JSPs, and Web Applications
- Security
- System Administration

## ClassLoader

---

<b>Change Request Number</b>	<b>Description</b>
077067	Setting the property <code>PreferWebInfClasses</code> in the <code>config.xml</code> now works correctly for resource files; if set, the files will now be taken from the users Web Application.
077344	When a class file was in a JAR file in the <code>WEB-INF\lib</code> folder, the <code>java.lang.ClassNotFoundException: org.apache.commons.logging.impl.LogFactoryImpl</code> was being thrown. This problem has been fixed.
079738	Fixed potential memory leaks when a web application with static references to a large memory structure is deployed or undeployed.
081377	Fixed a problem that resulted when classes were loaded via the context class loader at deployment time.
083752	If a class tried to create an <code>InitialContext</code> and the java command line specified <code>weblogic.jar</code> using the <code>-Xbootclasspath</code> argument, a <code>NullPointerException</code> was being thrown from <code>weblogic.i18ntools.L10nLookup.loadProps</code> . This has been fixed.
084236	The example <code>samples.examples.iiop.wls2wls.package-summary.html</code> now contains correct URL information.

---

## Cluster

---

<b>Change Request Number</b>	<b>Description</b>
CR043366	Comma delimited ip addresses can now be specified for the cluster address when the server is started in development mode.

---

CR078455,  
CR081750

In WebLogic Server 6.1 SP02 and SP03, a WebLogic Server cluster hung in session replication. The configuration included the F5 load balancer, Internet Information Server, and WebLogic Server ISAPI plug-in.

Thread dumps referred to a never-before-reported path in the RJVM subsystem.

```
ExecuteThread: '33' for queue: 'default' daemon prio=5
tid=0x49edf0 nid=0x2b waiting for monitor entry
[0xd527f000..0xd527fc68]^M at weblogic.rjvm.RJVMManager.
findOrCreateRemoteInternal(RJVMManager.java:213)^M at
weblogic.rjvm.RJVMManager.findOrCreate(RJVMManager.java:188)
^M at weblogic.rjvm.RJVMFinder.findOrCreateRemoteServer
(RJVMFinder.java:178)^M at weblogic.rjvm.RJVMFinder.
findOrCreate(RJVMFinder.java:149)^M at weblogic.rjvm.
ServerURL.findOrCreateRJVM(ServerURL.java:207)^M at
weblogic.jndi.WLInitialContextFactoryDelegate.getInitialCont
ext(WLInitialContextFactoryDelegate.java:311)^M at
weblogic.jndi.Environment.getContext(Environment.java:156)^M
at weblogic.jndi.Environment.getInitialContext(
Environment.java:139)^M at weblogic.servlet.internal.
HttpServer.lookupROIDS(HttpServer.java:751)^M at...
```

Analysis revealed that the ISAPI plugin directed requests to a server other than that primary or secondary. Then calls were made on the remote `ROIDImpl`.

This problem was resolved with a code fix in WebLogic Server 6.1 SP04.

---

CR078677

`HttpClusterServlet` now performs failover during the graceful shutdown of the managed server.

---

CR081908

Fixed a problem with HTTP session replication when the secondary server is unavailable at the TCP level. This solves a long delay when the server tries to connect to a secondary server that can obviously not receive a TCP request and has already been removed from the cluster view.

---

CR083532

Fixed a Null Pointer Exception at `weblogic.rmi.cluster.ReplicaAwareRemoteRef.getCurrentReplica` that was causing the server's shutdown sequence to be aborted prematurely.

---

CR087870

When using high-availability multi-pools, Weblogic now correctly fails over upon hardware failure when the pools have been created with an initial capacity equal to zero. The `ResourceException` that was being thrown was not the type that the multipool interpreted as reason to fail over--that was the problem.

---

## Console

Change Request Number	Description
057307	There is no longer an <code>NullPointerException</code> thrown when an attempt to change user password in the RDBMS realm is made.
061975	Fixed a problem in <code>createColumnsSql</code> when auto key gen is used.
073536	It is now possible to choose optional two-way SSL in the console by checking the check box.
074998	Fixed a problem receiving authentication errors when using form-based authentication and multiple locales. Credentials in languages other than English could be successfully added to the <code>fileRealm.properties</code> file but could not be used to authenticate via the servlet form-based authentication. The credentials can now be used to authenticate via the servlet form-based authentication.
079130	It is now possible to browse the JNDI tree for EJBs that are Deployed on the Managed Server.
079805	Now Web Applications that are targeted to the source server show up as targeted to the cloned server.
082578	Added a WAP-Enabled flag on the server config->http tab.
083107	The console now supports Mozilla 1.0 browsers.
083330	It is now possible to target a Multipool to a cluster via the console.
083377	If you specify each cluster node with a DNS name at <code>WebLogicCluster</code> , <code>HttpClusterServlet</code> is now able to maintain the stickiness of HTTP requests with references to the static server list.
083578	Increased the text field size for max message size on the server protocols tab.
084114	When trying to monitor the web applications in the console via the <code>mydomain --&gt; Servers --&gt; myserver --&gt; Deployments tab --&gt; WebApplications tab --&gt; monitor all active Web Applications</code> link, an exception was being thrown. This has been fixed.
086652	Now if you create a new domain within the console, the file realm tab will work in the left hand pane of the console.

## EJB

Change Request Number	Description
069899	Many-to-many CMR no longer throws an SQL exception if both tables have the same column as the foreign key.
076112	When a stateless session bean is deployed with the <code>&lt;max-beans-in-free-pool&gt;</code> parameter in the <code>weblogic-ejb-jar.xml</code> file, this value is not respected when concurrent calls are made to stateless session bean. WebLogic Server was creating further new instances of stateless session beans than the value specified in the parameter. This has been fixed.
076597	The <code>ejbRemove()</code> method was never being called when undeploying a stateless EJB with <code>&lt;max-beans-in-free-pool&gt;</code> to 0. This has been fixed.
077049	If an application uses dynamic proxies as parameters in EJB methods, a remote client can now use the interface even when the parameter interfaces are put on the server classpath.
078456	The client and server output that is documented in some of the EJB examples was difficult to read because the text was not wrapping. Now, the text does not wrap and is easier to read.
079745	EJBC no longer fails to deploy when another XML file is included in the <code>weblogic-ejb-jar.xml</code> file.
080569	After running ECperf, the <code>EJBCacheRuntime</code> statistics appeared with the <code>CachedBeansCurrentCount</code> on one stateful session bean (CartSes) printing out an incorrect (and negative) number. This has been fixed.
081355	EJBC no longer throws a <code>NullPointerException</code> when traversing relationships in an EJB 2.0 JAR file.
081807	EJB Deployer hard wired the <code>replicateBindings</code> to 'true' on context object before binding home to the JNDI tree so that even if the user set the <code>home-is-clusterable</code> to 'false' in the <code>weblogic-ejb-jar.xml</code> file, announcements to the rest of the nodes in the cluster are still being attempted a <code>NameAlreadyBoundException</code> is being thrown. This has been fixed.
081817	Provided an option to set <code>dispatchPolicy</code> at bean level in the <code>weblogic-ejb-jar.xml</code> file.

## 6 Resolved Problems

---

081991	A Message Driven Bean no longer fails to reconnect to MQSeries after “inhibited” and then “allowed” is selected for getting messages from the queue.
082029	Fixed a problem using EJB’s and connecting to a database with an XA driver when using an <code>openString</code> property for the connection pool.
082451	There is no longer a <code>ClassNotFoundException</code> thrown for third party JAR files for a Message Driven Bean that has a JMS provider that is not WebLogic Server.
083050	When calling <code>context.getCallerPrincipal()</code> from <code>ejbRemove()</code> , an exception is no longer thrown.
083098	The number of connections to MQSeries increases when the Queue is closed.
083689	CMP Entity beans with automatic key generation using <code>SQLServer</code> no longer results in a <code>javax.ejb.NoSuchEntityException</code> .
083896	When setting the <code>max-beans-in-cache</code> to 10 and then running the client to access the bean with the same primary key 200 times get 200 beans in the cache, if <code>concurrency-strategy</code> was set to <code>exclusive</code> , the database cache was ignored. This has been fixed.
083896	Fixed problems where the property <code>max-beans-in-cache</code> was being ignored.
084224	The <code>invalidate()</code> method was not being called by the container when a Read-mostly pattern was being used.
084978	You can now add and change EJBC parameters from the Administration Console. This allows you, for example, to change the amount of memory allocated to stub compilation.
085320	Fixed a problem with stateless session beans and the server; the ‘idle beans count’ displayed in the Administration Console was incorrectly continually increasing.
085659	Fixed some problems and delays with EJB failover when a machine is disconnected from the network.

## Examples

Change Request Number	Description
053356	The example <code>samples/examples/wtc/corba/simpappcns</code> did not have a <code>run.sh</code> UNIX script and did not work on UNIX machines. This problem was solved by providing <code>ubbdomain</code> , <code>dom1config</code> , and <code>run.sh</code> files, and instruction in the javadoc on how to build the tuxedo side. This solution provides multi-platform/machine support. The old build script assumed a single machine environment.
079543	The example server did not start when <code>BEAHOME</code> had a space in the name. This has been fixed.
084236	The example <code>samples.examples.iiop.wls2wls.package-summary.html</code> now contains incorrect URL information.
084903	The WTC <code>simpappcns</code> example has been fixed to work over the network and not just on a single machine.

## JDBC and jDriver

Change Request Number	Description
077974	Added OS authentication for Solaris because the OCI bug is fixed in Oracle 8.1.7 OCI.
080487	The JTS driver no longer leaks a pool connection if the DBMS rollback fails.
CR080931	Fixed an Oracle jDriver problem in which, when the <code>weblogic.oci.min_bind_size</code> property was set for the JDBC connection, a SQL UPDATE prepared statement issued after using <code>CallableStatement</code> on the same connection resulted in failure.
082438	Whereas statically defined pools absorb a property to define the prepared statement cache size, a pool created dynamically, via the console, <code>weblogic.Admin</code> , or an API was unable to do so. This has been fixed.

## 6 *Resolved Problems*

---

082484	The JDriver for Oracle no longer raises an <code>ArrayIndexOutOfBoundsException</code> if more than 511 parameters in a <code>PreparedStatement</code> are set.
086557	We have included the windows, HP, and Solaris jDrivers for Oracle 9.2 and updated the 8.1.7 and 9.0.1 drivers as well.
087005	A thin driver for Oracle 9.2.0 has been added/
087803	Fixed problems creating a JDBC store configured for a JMS Server using Oracle 9.0.1; the updated thin driver for 9.0.1 has been provided and solves this issue.
088156	Fixed a potential JDBC deadlock condition caused by the EJB persistence manager.

---

# JMS

---

<b>Change Request Number</b>	<b>Description</b>
063743	Fixed a problem that was resulting in Message Driven Beans not acknowledging object messages
079547	Fixed a deadlock that created problems while creating Durable Subscribers to JMS servers deployed in a clustered environment.
081525	Calling <code>TopicSubscriber.close()</code> in a cluster no longer throws a <code>weblogic.management.NoAccessRuntimeException</code> .
080668	In a clustered environment that sets listeners for the JMS Server, a <code>NoSuchObjectException</code> was being thrown if one of the nodes of the server was started after some time, rather than immediately. This has been fixed.
082298	The number of durable JMS subscribers was incorrect. An extra increment was occurring each time a consumer attached to a durable subscription. The number is now incremented when the subscribers are created and decremented when they are deleted.
082438	Prior to Service Pack 4, for JDBC connection pools that were created dynamically, the <code>STATEMENT_CACHE_SIZE</code> parameter was not implemented. This is corrected in SP04, so that dynamic connection pools can use the Statement Cache to cache and reuse prepared and callable statements.
083290	Fixed problems with JMS recovery in managed servers when the primary server goes down and then is restarted.
083503	Fixed a problem with JMS and Message-Driven-Bean recovery when starting and stopping managed servers.
084175	Messages were being lost when sent through <code>MessagingBridge</code> and interrupted by a server re-start. If while messages were being processed by the <code>MessagingBridge</code> (messages being processed from MQ Queue to WebLogic JMS Queue), and the WebLogic server was stopped and then re-started, then in some cases, some messages were lost. The number of messages lost corresponded to the <code>BatchSize</code> of the <code>MessagingBridge</code> . This problem has been fixed.
084182	The <code>setExceptionHandler</code> method no longer throws a <code>NullPointerException</code> when <code>exceptionListener</code> is set to null.

---

## 6 *Resolved Problems*

---

---

CR084374	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-24.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-24.jsp</a> .
CR086487	see CR83503
CR093060	Duplicate of “CR084374” above.

---

## Miscellaneous

Change Request Number	Description
049340	The <code>DeploymentOrder</code> on the <code>ComponentMBeans</code> can now be initially set based on the ordering in the <code>application.xml</code> file making re-deployment consistent with initial deployment.
053054	Fixed a problem where the server was not copying the transaction context. There were <code>rjvm</code> errors when a client that spawned a new JVM after binding an implementation class into the JNDI tree; the new JVM was not able to make RMI calls on the method.
053957, 071626, 075394, 076265, 078527, 079652, 079655, 082693, 082830	<p>This Service Pack improves WebLogic's handling of RJVMs in multiple ways. Fixes include:</p> <p>WebLogic Server now ensures that a valid connection has been fully established between servers running on local and remote Java virtual machines before it attempts to send a message to an RJVM.</p> <p>Possible deadlocks resulting from establishing a valid connection between servers running on local and remote Java virtual machines have now been fixed.</p> <p>Previously, it was possible for RJVMs to get into an infinite-wait state while waiting for the connection to be established because the notification of a successful connection was issued before the RJVMs went into a wait state. The fix makes it possible for a success notification to be detected, even if it happens before a wait.</p> <p>Multiple threads trying to establish a connection to an RJVM at the same time now rely on the success or failure of the first thread that actually manages to establish the connection.</p> <p>Previously, each thread would always attempt to establish a connection in turn, rather than using the connection established by the first thread that succeeded.</p> <p>Peer WebLogic Servers running on different computers now correctly detect when a peer server with which it has previously established a connection has gone, and correctly cleans up the broken connection.</p> <p>The Remote Method Invocation (RMI) layer now receives proper exceptions to failover when an attempt to get an <code>OutputStream</code> fails.</p> <p>Previously, when two peer servers were trying to establish a connection but failed to, an exception was not thrown to the RMI layer, causing improper failovers or delays to a proper failover.</p> <p>Possible deadlocks have been fixed between 2-way calls from <code>T3JVMConnection</code> to <code>ConnectionManager</code> to RJVM.</p>

## 6 Resolved Problems

---

---

056403, 063910, 076409, 076903, 078288, 078918, 079599, 080292, 080744, 081010, 081116, 081856, 082700,083623, 086362.	<p>We fixed multiple muxer problems and added improvements including:</p> <p>Collections in all muxers (Java and native-code) that map to sockets now have unique keys. This has solved a variety of problems, including data corruption, bad pointer references, WebLogic Server hangings and crashes.</p> <p>Data structures and state machines in muxers now have proper synchronization. This fixed certain classic deadlocks caused by conflicting order of locking.</p> <p>The call stacks between the muxer and the protocol layers above the muxer are now 2-way, which means a close of a socket or connection can be initiated from any point in the stack. Safeguards have been added to prevent deadlocks.</p> <p>State machine transitions in the muxer have been fixed, resulting in sockets no longer remaining in a CLOSE_WAIT state.</p>
064301	<p>If the application server is killed abruptly, there is a high probability of creating a distributed in-doubt transaction. In WebLogic Server 6.1 SP01, the distributed in-doubt transaction still continued to exist and needed to be resolved by manual DBA intervention. In WebLogic Server 6.1 service pack 4, the transaction log file is re-scanned and a recovery process is initiated after re-starting the application server.</p>
071510	<p>It was not possible to start the server using JMX through the <code>NodeManagerMBean</code>. In WebLogic Server 6.1 service pack 4, new methods were added in the <code>ServerMBean</code> to start and kill <code>ManagedServers</code> via the JMX client.</p>
072188	<p>The admin server was not closing the network connection that it established with <code>NodeManager</code> after completing the task. This has been fixed.</p>
073023	<p>Node manager no longer randomly throws the <code>OutputHandler</code> error when trying to start a remote server.</p>
074844	<p>Fixed a problem that was causing an assertion error when using dynamic proxies.</p>
075406, 076002	<p>Fixed problems occurring under heavy load where <code>NullPointerException</code> and <code>ClassCast</code> exceptions were occurring; also made some improvements to the NT muxer.</p>
076605	<p>Fixed a problem that was causing the message: <code>&lt;ExecuteThread: '12' for queue: 'default' &gt; &lt;&gt; &lt;&gt; &lt;000000&gt; &lt;No IORecord present for fd&gt;</code> to appear.</p>
076705	<p>Message Driven Bean deployment no longer fails and results in a <code>NoAccessRuntimeException</code> for guest if the destination is remote.</p>
076997	<p>Now, when deploying an EAR component to a cluster through MBeans, the JNDI bindings appear in all managed servers without the need to re-start the servers.</p>
077248	<p>Fixed problems connecting to EJB's over IIOP when native IO is turned off.</p>

---

077831	When the AppletArchiver utility was used to generate a client JAR file and if the applet in question used the <code>javax.swing.JApplet</code> class in its code, then the AppletViewer was producing an error. This has been fixed.
077919	Using patch CR064447_61sp2.jar was causing an error in the log file after calling <code>ObjectOutputStream.writeObject()</code> on <code>EJBHandle</code> . The output of the <code>EJBObject</code> in the log was changing from <code>LocalServerRef</code> to <code>LeasedRemoteRef</code> . This has been fixed.
078431	Fixed a problem that was causing a deadlock between <code>MuxableSocketHTTP</code> and <code>PosixSocketMuxer</code> .
078952	An <code>org.omg.CORBA.MARSHAL</code> exception is no longer thrown while a null value in a serializable object from an EJB's remote method is being returned.
079220	When running JMS, the client was getting the exception: Could not find RJVM for \$Proxy68.. (IllegalArgumentException) in a cluster environment. The client was contacting one of the managed servers to drive. This has been fixed
079672	A VM crash is possible when <code>WarClassFinder.getSource</code> is invoked. Sun's fix for this problem are in JDK's 1.4.1_02 and 1.3.1_07. The Sun bug is: <a href="http://developer.java.sun.com/developer/bugParade/bugs/4369396.html">http://developer.java.sun.com/developer/bugParade/bugs/4369396.html</a> .
080177	Using <code>weblogic.deploy</code> no longer causes memory leaks if you continually deploy and delete the same EJBs.
080740	If a web application that creates a file that got deleted when the server was shutdown, when running as a Windows Service, this file got deleted as expected when the server was shutdown from the command line (using the <code>java weblogic.Admin SHUTDOWN</code> command) or from the administration console. However, this file did not get deleted when the server was shut down by going to: Start Menu --> Settings --> Control Panel --> Services --> Press Stop button for the service. This capability is now implemented.
080895	When a stand-alone java client was calling a stateless session EJB, after about 500000 iterations, the following exception was being thrown on the client side: Exception in thread "main" java.lang.ClassCastException: Assigning instance of class java.io.ObjectStreamClass to field <code>weblogic.rmi.internal.ClientMethodDescriptor#signature</code> at <code>java.io.ObjectInputStream.inputClassFields(ObjectInputStream</code> <code>.java:2271)</code> This has been fixed.

## 6 Resolved Problems

---

080901	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-19.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-19.jsp</a> .
080929	Now, if using wildcards, <code>weblogic.refresh</code> can refresh files.
081311	A <code>SAXParseException</code> on the DTD now has a severity of <code>&lt;Warn&gt;</code> instead of <code>&lt;Info&gt;</code> .
081404	The JMS Bridge Resource Connectors ( <code>jms-xa-adp.rar</code> , <code>jms-notran-adp.rar</code> , <code>jms-notan-adp51.jar</code> ) for Messaging Bridge are packaged with WebLogic Server 6.1 Service Pack 4.
081568	SNMP Agent can now monitor the <code>ServletRuntime</code> of Servlets.
081732	Using WebLogic Server 6.1 SP03, an exception was being thrown when using the Sybase database while trying to start the server. This has been fixed.
081765	There were problems with SSL closing sockets prematurely and causing muxer errors.
081853	Fixed some complicated problems with RMI-IIOP and the marshalling of complex objects, notably in environments using a mixture of JDKs.
081868	In WebLogic Server 6.1 service pack 2, when the admin server was shutdown using <code>java weblogic.Admin SHUTDOWN</code> , the SNMP agent was not generating the <code>serverShutDown</code> trap. It was able to generate <code>serverStart</code> trap when the server started up. Now, when the admin server is shutdown, the SNMP Agent will issue a self shutdown trap.

---

---

CR081870	<p>In the WebLogic Server distribution of Xalan, when the XSLT output method was set to HTML (required):</p> <pre>&lt;xsl:output method="html" /&gt;</pre> <p>the spaces in HREF attributes in anchor tags are encoded to "%20". For example:</p> <pre>&lt;a href="javascript: var win = openWindow()"&gt;Link&lt;/a&gt;</pre> <p>erroneously outputs as:</p> <pre>&lt;a href="javascript:%20var%20win%20=%20openWindow()"&gt;Link&lt;/a&gt;</pre> <p>Netscape fails to interpret the "%20" as a space, breaking the link. If the output type is changed to "XML" (not an applicable workaround in this case), the attribute outputs correctly.</p> <p>The vanilla Xalan distribution (i.e. not the BEA-enhanced version) performs the transformation correctly with identical input.</p> <p>WebLogic Server distribution of Xalan performed inappropriate URL encoding on the HREF attribute. This works fine for normal URLs (i.e., http://somehost/...) where spaces are not allowed, however javascript URLs (i.e. javascript: doSomething()) are also valid and should not be encoded. The XSLT transformer should not be doing anything to the URL as there are already ways to explicitly encode URL's in stylesheets.</p> <p>Problem occurred in WebLogic Server 6.1 SP2, SP3, and SP4.</p> <p>Problem was solved by modifying the XSLT transformer to not encode the URL. .</p>
082004	<p>The CR073910_61sp2.jar patch on WebLogic Server 6.1 service pack 2 was resulting in a memory leak. when getting a new <code>InitialContext()</code>.</p>
082263	<p>The <code>weblogic.deploy</code> tool is now able to update the web application .war file from Win2k to a server running on Solaris.</p>
082533	<p>Debug messages were being written to <code>std.out</code> when using the Xerces Parser. These messages have been removed.</p>
082693	<p>In WebLogic Server 6.1 service pack 1 and 2, if the server received a call from the old generation stubs, the server correctly sent <code>peerGone</code> to the client and logged an error. In WebLogic Server 6.1 service pack 3, the server printed a <code>NullPointerException</code> and the client waited until timeout. Now we have restored the earlier, correct, behavior.</p>
083102	<p>When deploying a couple of EJBs (that conform to the EJB 2.0 specification) with <code>J2EE12OnlyModeEnabled="true"</code>, the server was throwing a couple of warning messages for a few beans and then crashing with a <code>DistributedManagementException</code>. This has been fixed.</p>
083485	<p>When invoking a t3 client, the client no longer hangs in <code>JavaSocketMuxer</code> with t3 as the default protocol.</p>

---

## 6 Resolved Problems

---

083652	In WebLogic Server 6.1 service pack 3, applications were undeployed at startup and then redeployed twice. This has been fixed.
083654	Improved the performance of WebLogic Server 6.1 service pack 3, when continuous requests are issued from the Microsoft WAS tool. International users, see “089868” on page 5-33 for related information.
083721	The <code>FailureIsFatal</code> option of the startup class no longer suppresses the runtime exception on a Solaris Environment.
083920	Suppressed any startup traps when the admin server is re-started and it discovers that managed servers are up to avoid redundant traps being sent when the admin server is re-started.
084127	In production mode, even with auto-deployment turned off, WebLogic Server was still validating some application files that were not configured in the <code>config.xml</code> file. This has been fixed.
084622	Using ZAC no longer results in an assertion error when <code>getInitialContext()</code> is called.
085463	JNDI lookup failover now no longer takes such a long time when the network plug is pulled.
085914	SNMP request was timing out after a managed server was dropped. The SNMP agent no longer becomes unresponsive for a while on <code>getNext</code> requests when a managed server is down.
085979	Fixed a problem with <code>MBeanHome.getAllMBeans()</code> unnecessarily throwing an exception when a custom Mbean is defined.
086108	We improved performance; the <code>ListenThread</code> now consumes less memory than in Service Pack three.
086248	The component type for boxed RMI sequences was being forwardly declared twice—once correctly and once as an interface; the wrong type for sequences of valuetypes.
086350	Fixed a problem where if an exception’s classname ended in a capital ‘T’ the IDL generated from it is mangled and corrupt.

CR086362 Under heavy loads SocketException: Connection timed out errors occurred. Stack trace:

```
#####<Sep 18, 2002 9:37:07 AM EDT> <Error> <Posix Performance Pack> <lpsapp01> <LPSAPP01_C> <ExecuteThread: '34' for queue: 'default'> <> <> <000000> <IOException on socket: 'weblogic.rjvm.t3.T3JVMConnection@67cf01', fd: '1046'>
java.net.SocketException: Connection timed out: Connection timed out
Start server side stack trace:
java.net.SocketException: Connection timed out: Connection timed out
at java.net.SocketInputStream.socketRead(Native Method) at
java.net.SocketInputStream.read(SocketInputStream.java:86)
at
weblogic.socket.PosixSocketMuxer.readBytesProblem(PosixSocketMuxer.java:824) at weblogic.socket.PosixSocketMuxer.deliverGoodNews(PosixSocketMuxer.java:712) at weblogic.socket.PosixSocketMuxer.processSockets(PosixSocketMuxer.java:637) at weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:24) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) End server side stack trace
and the server instance hung with sockets in the CLOSE_WAIT state. Thread dump:
"ExecuteThread: '3' for queue: 'default'" daemon prio=5
tid=0x7f79a8 nid=0x10 waiting for monitor entry
[0xa6101000..0xa6101a40] at weblogic.socket.PosixSocketMuxer.read(PosixSocketMuxer.java:542) at weblogic.socket.JVMSocketManager.accept(JVMSocketManager.java:185) at weblogic.t3.srvr.ListenThread$RJVMListenRequest.execute(ListenThread.java:594) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
Thread dump analysis indicated that the thread in native AddFdToPollSet did not finish writing to PollSet Loopback FD, and release POLLSET_LOCK properly.
Problem was resolved by tracking the number of bytes written to pipe, and waiting to write more to pipe (using an array instead) until thread reader notification that it is OK to write to the pipe.
```

---

086761 The `Weblogic.admin` tool now correctly creates an execute queue so that it will be functional on server start-up.

---

087265 Fixed several incompatible serialVersionUID's so as to keep interoperability between WebLogic versions.

---

## 6 *Resolved Problems*

---

---

088372	Now, under load, we have fixed the problem that was generating multiple exceptions like: <code>ArrayIndexOutOfBoundsException</code> in <code>weblogic.security.acl.TTLCache.cleanup</code> .
089044	Loading the Administration Console no longer causes a browser/OS compatibility check.

---

## Plug-ins

Change Request Number	Description
074128	<p>In a cluster, after Internet Explorer 10053 displayed a socket error message, the proxy plug-in inappropriately failed over—the primary server was healthy. The problem was exhibited in this configuration:</p> <ul style="list-style-type: none"> <li>■ WebLogic Server 6.1.0 with SP02</li> <li>■ IIS plugin</li> <li>■ Windows 2000 SP2</li> <li>■ IIS 5.0</li> </ul> <p>Error 10053 is a winsock error and is defined in <code>winsock.h</code> as <code>WSAECONNABORTED</code>. The error occurs when a page is reloaded before the ISAPI extension is done sending data back. When the page is reloaded, the browser closes the existing connection and then creates a new connection to resubmit the request. Same condition can be caused by using the "Stop" button in the browser before the extension is done sending data.</p> <p>This is a normal event that means that the browser isn't listening any more. After the error, plug-in should simply clean up and return from <code>HttpExtensionProc</code>.</p> <p>The inappropriate failover case has been solved by changes to the IIS plugin. Failover no longer occurs when processing fails during the <code>sendRequest</code> phase if timeout occurs reading POST data.</p>
076936	<p>When specifying a multiple-level path at <code>PathTrim</code> under the ISAPI environment using <code>WlForwardPath</code> together such as:</p> <pre>WlForwardPath=/ PathTrim=/path/to/weblogic</pre> <p>ISAPI plug-in was not able to trim the specified path. This has been fixed.</p>
078713	<p>Added a new parameter set <code>FilterPriorityLevel</code> in the <code>iisforward.ini</code> file. The value for this parameter is 0, 1, and 2, which means low, medium and high for the filter priority. The default value is 2. If it is not possible to use a virtual host to set this property, use the <code>iisforward.ini</code> file.</p>
079186	<p>In accordance with the HTTP 1.1 specification, all plug-ins will now parse folded headers properly.</p>
079683	<p>Chunked Transfer Encoding HEX numbers were displayed to the browser when using <code>mod_wl_ssl.so</code>. This only happened when using <code>jsp:include</code> on Weblogic. The <code>mod_wl.so</code> worked correctly, but there was a long delay before the browser rendered the page completely. This has been fixed.</p>

## 6 Resolved Problems

---

079973	In WebLogic Server 6.1 service pack 4, if <code>Idempotent</code> is off, failover is prevented and if <code>ConnectTimeoutSecs=0</code> , core dump is prevented. Retries are no longer attempted after <code>ConnecyTimeoutSecs</code> is set to 0 in the iPlanet config. The retries were attempted either with the default setting, which is 10, or with 0. Furthermore, when <code>Idempotent</code> was set to ON, the retries occurred five times, instead of two when it was set to OFF.
080219	There is no longer an iPlanet core dump when shutting down all WebLogic Server instances.
080382	An unexpected error message is no longer logged in the WebLogic Server error log when using the NSAPI plug-in.
080746	A 64-bit platform on native webserver plug-in is now supported.
081185	The CPU load no longer increases when cancelling the uploading of files.
081493	Fixed a problem that was resulting in HPUX-11 iPlanet crashing.
082093	When WebLogic Server is being hit via an iPlanet proxy with an NSAPI plug-in and a QALoad is used to generate a number of concurrent client connections, extra calls are no longer being made by the proxy plug-in to the app server under a heavy load.
082096 & 083386	With <code>Idempotent=OFF</code> , the IIS proxy plug-in no longer still attempts one re-try (two requests = one original + one re-try) for all POST requests after <code>HungServerRecoverSecs</code> .
082113	If the plug-in receives an error while reading data from the client, it no longer tries to send the incomplete data (request) to the back-end WebLogic Server.
082206	You can now exclude certain pages from being forwarded to WebLogic and have them be served by an IPlanet front-end. Add <code>WLExcludePathOrMimeType</code> to your <code>obj.conf</code> to exclude certain requests that match with the expressions you define (i.e: <code>WLExcludePathOrMimeType="/test,/*.html"</code> )
083174	Using the latest IIS plugin, failover was not working when new values were being added to the session.
083643	Fixed a performance problem in NSAPI. The <code>netbuf_getc</code> function was very slow and not able to distinguish between <code>'\0'</code> and <code>0</code> in the postdata.
083558	Certified the newest Apache-Apache 2.0.42.

## Servlets, JSPs, and Web Applications

Change Request Number	Description
042655	When using the <code>access.log</code> in the extended log format, the times at which the http requests are made are no longer given as the local time, instead of the GMT time.
053974	Besides being able to configure the name of the <code>access.log</code> , users can now select the convention for the rotated log for manageability.
058389	When <code>getInputStream()</code> is called against <code>javax.servlet.http.HttpServletRequestWrapper</code> , it no longer returns null.
063630	If a Web Application failed to deploy, then no other applications were getting deployed to a cluster. This problem has been fixed.
065967	Taglibs no longer fail when <code>&lt;type&gt;</code> is specified and the type is an array. Weblogic Server no longer sees it as a string when it is actually an array.
068577	In the generated code, the method <code>_releaseTags()</code> was not being called upon a <code>&lt;jsp:forward&gt;</code> although according to the JSP specification 1.2, all tag handlers need to be released upon <code>DO_END_TAG</code> . This has been fixed.
069731	When using JSPs in a webapp, the <code>.java</code> files that get created from the JSPs live under <code>_tmp_war</code> (under the webapp root). If it does not exist already, the <code>_tmp_war</code> directory is created at the time of WebApp deployment (during the initialization period after you start the server). If the server is run as root, the <code>_tmp_war</code> directory gets created by, and is owned by, root. At the end of initialization, <code>nonPrivUser</code> was being used to switch to a nonprivileged user and the <code>_tmp_war</code> directory was inaccessible by the JSP servlet and WebLogic. This has been fixed.
071513	WebLogic Server is now validating the <code>weblogic.xml</code> file in the webapp.
073780	Improved the performance of JSP tags when using WebLogic Server 6.1 service pack 2 and the following three patches: <code>CR065452A_610sp2.jar</code> , <code>CR063457A_610sp2.jar</code> , and <code>CR063829_610sp2.jar</code>
073791	Fixed problems were customers were getting a <code>NullPointerException</code> in the <code>ChunkUtils.getEnd</code> method after calling <code>ServletOutputStreamImpl.write</code> from their servlet code.

## 6 Resolved Problems

---

074265	The <code>ServerMBean</code> has a new attribute called <code>MaxBackoffBetweenFailures</code> . The default is 10 seconds. Set this to change the time period in which the server listen thread will send a "Failed to listen on port" if it is still trying to accept the socket connection.
074940	A <code>NullPointerException</code> is no longer thrown when using the <code>size</code> Cache Tag Attribute of the Cache Tag of the <code>weblogic-tags.jar</code> .
075471	JRun tag libraries are working on WebLogic Server service pack 4.
075721	The <code>jsp:useBean</code> tag with the <code>beanName</code> of <code>TimeExpression</code> is no longer interpreted incorrectly.
076375	<p><code>weblogic.log</code> and <code>access.log</code> (HTTP Logs) will now be renamed based on the configuration of the log filename. Usage of % symbols in the logfile name will invoke the logic which renames log files based on the date/time when it gets rotated. The file being currently logged into will be the log filename stripped off its % symbols if they exist. Examples of valid formats:</p> <pre>weblogic_%yyyyMMdd%_%HH%_%mm%.log foobar_%yyyy%-MM%-dd%_%HH%_%mm%.log</pre> <p><b>Note:</b> Y (capital) does not represent anything in the <code>SimpleDateFormat</code>. Use lower case y (small) to represent the year. Invalid formats will fall back to WebLogic Server default log file names. A string enclosed between percentage signs (%) will be decoded based on <code>SimpleDateFormat</code>. This format applies to the filename, not the directories.</p>
077018	Prevented a <code>ClassCastException</code> in <code>ServletAuthentication</code> when using request/response wrappers.
077306	Temporary patches no longer cause recompilation of included (precompiled) JSP classes.
077422	The <code>CookieParser.cookieToString</code> creates the <code>expires</code> cookie with a <code>SimpleDateFormat</code> that contains a comma, ','. The comma in the <code>expires</code> date is causing the cookie to be parsed incorrectly when it is returned in the request. The comma is interpreted as a cookie separator and breaks the parse of the <code>expires</code> date. This has been fixed.
077888	<code>setCharacterEncoding</code> will now throw <code>UnsupportedEncodingException</code> .
078090	According to the JSP specification, if you use <code>jsp_precompile</code> in the request parameter, the container should compile the JSP (if not compiled), but must not display the JSP. Since WebLogic Server parses the query string and finds the <code>jsp_precompile</code> , it was not displaying the page on the browser.
078262	Added <code>getResourcePaths(String)</code> to the <code>ServletContext</code> .

078325	Fixed a problem that was resulting in an <code>UnsupportedEncodingException</code> if there were extra quotes around the charset value.
078698	The <code>access.log</code> file is now rotating by size when the log size exceeds 2MB.
079582	WebLogic Server 6.1 service pack 3 threw a <code>SocketException</code> during data sync. This has been fixed.
079767	When redeploying EAR, WAR, or JAR files to WebLogic Server, with every redeployment of an application or component, the <code>.wlnotdelete</code> directory of the managed server no longer increases in size.
079892	The CPU usage no longer pegs to 100% while executing a CGI script and hitting the stop button in the middle.
080090	When re-directing, the location header will always use the following parameters, if they are specified: <code>FrontendHTTPPort</code> <code>FrontEndHTTPSPort</code> <code>FrontEndHost</code>
080384	The <code>HttpClusterServlet</code> was marking a server as bad, even if just one response had been scrambled in the status code header. This has been fixed.
080751	Custom exceptions extending servlet exceptions will now be redirected to error page if configured.
080791	Folded headers in the cluster/proxy servlets are now considered.
080837	Added a new element in the <code>weblogic.xml</code> file: <pre>&lt;!-- The wl-dispatch-policy can be used to assign the webapp to a configured execute queue by identifying the execute queue name. This webapp level param can be overridden at the individual servlet/jsp level. eg: &lt;servlet&gt; ... &lt;init-param&gt; &lt;param-name&gt;wl-dispatch-policy&lt;/param-name&gt; &lt;param-value&gt;CriticalAppQueue&lt;/param-value&gt; &lt;/init-param&gt; &lt;/servlet&gt; --&gt; &lt;!ELEMENT wl-dispatch-policy (#PCDATA)&gt;</pre>

## 6 *Resolved Problems*

---

081071	There is no longer a <code>NullPointerException</code> thrown at <code>weblogic.servlet.internal.ServletContextManager.trimAbsUrl</code> .
081484	Session attributes are no longer lost when using the <code>HttpSessionListener</code> with <code>PersistentStoreType</code> as replicated but able to retrieve the session attributes when non-replicated.
081521	It is now possible to call a CGI script from a JSP.
082156	<code>isRequestedSessionIdValid()</code> always returned the value, <code>true</code> , which means that the request session ID is valid, even if you changed the session ID in the URL. Invalid sessions are now correctly detected and reported as invalid.
082238	<code>CreateSessionServlet</code> was checking for the existence of the <code>Cache-Control</code> header, setting it to a valid value if it exists, or adding it to the header list if it does not exist. In both cases, it added another bogus header. This problem was fixed by setting <code>CacheSessionCookie</code> to <code>true</code> in <code>weblogic.xml</code> .
082310	JSPs with a <code>page contentType</code> directive can now be deleted or overwritten in an exploded WAR file.
082580	HTTP POST parameters were not preserved if a form-based authentication is invoked and authentication is successful when accessing a URL in WebLogic Server 6.1 service pack 3.
082671	Made <code>HttpSession.getServletContext()</code> public in the internal implementation.
083191	The console is no longer displaying incorrect values for Servlet Average Execution Time.
083200	WebLogic Server is now filtering correctly with <code>nonProxyHosts</code> .
083487	<code>CGIServlet</code> now works with exact match <code>&lt;url-pattern&gt;</code> .
083517	<code>HttpClusterServlet</code> was not proxying a request to SSL even when <code>SecureProxy</code> was ON. This has been fixed.
083597	In WebLogic Server 6.1 service pack 3, if the content type using the page directive is set as: <pre>&lt;%@ page contentType="text/html; charset=UTF-8" %&gt;</pre> the content type was not set properly and the desired output was not obtained. This has been fixed.
083912	An ORA-01461 error no longer occurs while the server is trying to write the session data in the JDBC managed persistence when using the Oracle thin Driver.

---

084002	When WebLogic Server returned an HTTP code 204, it set the Content-length to 886 with an HTML error page. Therefore, the load runner client was failing and generating an error stating that "204 responses codes should not have a body but the Content-length was set to 886 (with an HTML error page)". This has been fixed.
084030	When an exploded web-app was shared by two servers, the generated tmp lib directory got deleted at the second server's start. The directory is no longer deleted.
084058	The <code>request.getUserPrincipal()</code> method was returning guest after <code>ServletAuthentication.logout()</code> . Now, the <code>getRemoteuser()</code> returns null.
084536	Fixed a problem where the patch for CR083377 was causing problems with security and the SSL connection.
084649	There are certain environment variables that a CGI server is expected to provide to scripts that run within it. <code>SERVER_URL</code> , <code>HTTP_COOKIE</code> , and <code>QUERY_STRING</code> , which are present in Netscape's CGI server, were unavailable when running the same CGI program inside of WebLogic's CGI servlet. When a request was made for one of these variables, NULL was returned instead of a value. This has been fixed.
084785	The <code>GenericProxyServlet</code> can now handle wrapped responses.
084847	For chunked transfer, WebLogic Server was including a hexadecimal number which other servlet engines used to ignore. This has been fixed.
085754	If newlines/comments were put in the intermediate JSP <code>BEATestCase2.jsp</code> , when used with a request dispatcher, the client got a blank page. There was no exception thrown on the server side. This has been fixed.
086026	The parameter, <code>CacheSessionCookie</code> , used to default to false. Now, it defaults to true.
086052	HTTPS connections outbound from WebLogic Server did not work without a trailing slash '/' on the URL. This has been fixed.
086280	There is no longer a <code>NullPointerException</code> thrown when accessing a servlet over a Keep alive connection.
086481	Fixed some problems when a web application downloads a larger file and the buffer overflows while reading the multi-part form data.

---

## 6 *Resolved Problems*

---

---

088301	<p>Fixed problem where an included resource was not sent to the servlet output stream when dispatched from another resource that was Forward dispatched inside a JSP BodyTag.</p> <p>NOTE: The nested BodyTag tag output stream was not being flushed since the forward dispatched request caused the tag to exit - but WebLogic still presumed the output should go to the Tag's nested output stream. This was resolved by clearing out reference to the BodyTag nested output stream when the Forward request was dispatched.</p>
CR089803	<p>Duplicate of “CR090225”.</p> <p>Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-14.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-14.jsp</a>.</p>

---

## Security

Change Request Number	Description
058355	When a user called <code>login()</code> on the <code>LoginContext</code> , this call was directly mapped to the <code>LoginModule</code> 's <code>login()</code> , but the <code>logout()</code> method was not implemented. Therefore, a call to this method could not be mapped to the <code>LoginModule</code> 's <code>logout()</code> . In WebLogic Server 6.1 service pack 4, <code>LoginModule.logout()</code> is implemented.
057950	For security reasons, the <code>Random String</code> field was removed from the <code>Certificate Request Generator</code> servlet.
068729	The <code>weblogic.security.SSL.SSLCertificate</code> class is now documented.
075109	<code>LDAPRealV2</code> now caches memberships.
075451	When trying to obtain an <code>InitialContext</code> to a remote WebLogic Server from within a WebLogic Server, after making an SSL connection with <code>weblogic.net.http.HTTPURLConnection</code> , an exception is no longer thrown.
076783	Using the RDBMS realm, extra calls were being made. The caching realm is now fixed to avoid having these extra calls made.
077233	The absolute path of SSL certificates is now given.
077288	In WebLogic Server 6.1 SP02, "=" is included in username when parsing CN for a user in a group. This problem was resolved by a correction to the lexing code in the <code>LDAPRealmv2</code> .
077425	When a custom realm was used with a caching realm and the case insensitive cache was set to a default of true, the caching realm was throwing an error message, "Server cannot check in a case-insensitive way". This has been fixed.
078893	If <code>weblogic.management.password</code> is not blank, the custom login module never got called. For <code>j_security_check</code> to use the JAAS loginmodule, it is necessary to use programatic authentication ( <code>servletauthentication.weak()</code> ) in combination with JAAS loginmodule.
079183	After calling <code>weblogic.security.acl.ManageableRealm.setName</code> , <code>weblogic.security.acl.ManageableRealm.getAcl</code> no longer returns null.
079364	<code>deleteAcl</code> was only working with the <code>getACLOwner</code> principal. This has been fixed.

## 6 *Resolved Problems*

---

---

079637	Added the pkpassfile property to specify the name of the file containing the private key password.
082003, 087419	Using an LDAP realm configured for Microsoft Active Directory Server, basic authentication was failing while form-based authentication worked fine. This has been fixed.
082098	Credentials are now used / verified against the remote server realm.
083478	Even though the email is not required in the certificate form, EMAIL= was appearing in the certificate. This has been fixed.
084166	Fixed the caching of delegate groups so that with a nested group there is improved server performance.
087485	With WebLogic Server 6.1 SP03, LDAP was calling <code>Group.isMember()</code> for every member of group. In this Service Pack we restored the original behavior of an LDAP realm: now if the <code>groupmembershipcache</code> value is set to 0 then WebLogic checks the LDAP realm for group membership, thus preventing loading all members() if the group contains many members.

---

## System Administration

Change Request Number	Description
059779	Data sources can now be bound to the same JNDI name on different servers.
062102	Applications are no longer copied to the managed servers upon startup if they do not need to be copied which results in increased disc space.
063630	It is now possible deploy remaining applications to the managed servers upon a failed deployment.
072833	When a managed server was configured from the admin console with the listenAddress field blank, WebLogic Server was allowing the multiple instances of this managed server with the same name to be run from different machines. This is no longer possible.
074370	When using WebLogic Server 6.1 service pack 2, a <code>javax.management.AttributeNotFoundException</code> was being thrown when using the EJB descriptor editor in the admin console. This has been fixed.
074653	The Admin console Deployment Descriptor Editor now replaces "&" with "&#" when writing to memory/disk because "&" is a special character which must be escaped as "&#" in XML files.
075949	Fixed a problem that was causing <code>weblogic.deploy</code> to undeploy the JAR file immediately after deployment.
076539	Web apps with <code>load-on-startup</code> are now loaded in the correct order according to <code>DeploymentOrder</code> .
078257	The admin server JNDI was not showing the managed servers' <code>MBeanHome</code> causing a <code>NameNotFoundException</code> to be thrown. This has been fixed.
079270	The application redeployment (delete & deploy) operation was not always successful when some of the targeted managed servers were down. For example, the JNDI entries for EJB were not registered in running servers, or "Targets" attributes in the <code>config.xml</code> file were lost because of a warning message. This has been fixed.
079455	A <code>ConfigurationException</code> is no longer thrown upon deployment failures.
079819	The time it takes managed servers to start up with a large number of deployed applications has been shortened.

## 6 Resolved Problems

---

080016	There is no longer an error deploying an application while starting the admin server if the application targeted to the virtual host (which has been targeted to managed server) and the admin server.
080324	Exposed the <code>setParent</code> method to a public API.
080690	If the admin server was bounced and re-started, it re-established a connection to the managed server, but the JNDI tree did not show the managed servers. As a result, the application was getting a <code>NameNotFoundException</code> . It would only show up in the admin JNDI tree once the managed servers were re-started. This has been fixed.
080778	Turning on session monitoring no longer causes traffic to the admin server every time a new session is created.
081736	After <code>weblogic.Admin -help</code> prompts for a guest password, it no longer ignores the password.
082765	When calling shutdown to shutdown a managed server, <code>removeManagedHome</code> is also called to remove the managed server from the cache. Otherwise, needless RMI calls would be made when any other managed server boots causing startup delays.
082910	It is now possible to remove a JDBC Connection Pool using MBeans when a customer MBean is registered from within an EAR file.
083400	There is no longer a <code>StackOverflow</code> during managed server startup using T3s as the default protocol.
084484	When starting a server from a directory other than the domain directory, the option <code>-Dweblogic.RootDirectory=&lt;root directory&gt;</code> can be specified. If the start script is changed to go to the root directory, it would find the application, but the <code>config.xml</code> path got modified to be an absolute path with the <code>RootDirectory</code> appended. The <code>config.xml</code> path no longer gets changed to an absolute path.
084740	When deploying an application on a cluster using <code>weblogic.deploy</code> , the deployment was not effective until the server was restarted.
086235	When trying to get the server version out of the <code>ServerRuntimeMBean.getWeblogicVersion</code> method, only the version info for the patch that was the first in the classpath was returned. In WebLogic Server 6.1 service pack 4, the whole version information is returned by the <code>weblogic.Admin</code> command.

## Web Services

---

<b>Change Request Number</b>	<b>Description</b>
076510	When there was a webservice that had a method that received a String as a parameter that is actually an XML document with CDATA sections, after invoking the method, the following response was sent: <code>weblogic.soap.S SoapFault: Connector - Bad request to the server.</code> This has been fixed.
082661	Now, wsgen handles an extra classpath.
084960	Fixed a problem using the <code>URLConnection</code> method to make an HTTP connection to a Web Service. Weblogic overrides <code>java.net.HttpURLConnection</code> with its own version which will throw an exception from the <code>getInputStream</code> method if a status code is greater than 400. When the exception is thrown there is no way for the WebService client to retrieve the input stream, and thus the HTTP SOAP Binding is broken. If you have this problem, set the new command line option <code>-Dweblogic.net.http.ignore500status</code> to true.
085214	Fixed a problem that occurred with webservices when he passing dates as parameters. A <code>java.lang.IllegalArgumentException</code> is no longer being thrown.

---

## WebLogic Tuxedo

---

<b>Change Request Number</b>	<b>Description</b>
077553	Integer array object passed by value from Tuxedo to WebLogic Server threw a <code>weblogic.utils.AssertionError</code> . The problem was resolved with a code fix.

---

## XML

<b>Change Request Number</b>	<b>Description</b>
080756	When converting Japanese characters by using the built-in XSLT processor, the characters got changed into the unexpected code. This has been fixed.
080961	Fixed a Xerces 1.3.1 bug in <code>org.xml.sax.helpers.AttributesImpl.removeAttribute()</code> .
081372	<code>DocumentBuilder.parse()</code> is no longer multi-thread unsafe when input encoding is 'Unicode'.
081870	When the XSLT output method was set to HTML (required): <code>&lt;xsl:output method="html" /&gt;</code> the spaces in HREF attributes in anchor tags were encoded to "%20". Netscape failed to interpret the "%20" as a space, breaking the link. This has been fixed.

# WebLogic Server 6.1 Service Pack 3 Solutions

The following sections describe problems that were resolved for the release of WebLogic Server 6.1 Service Pack 3.

- Classloader
- Cluster
- Console
- EJB
- Examples
- JDBC
- jDriver

- JMS
- JTA
- Miscellaneous
- Plug-Ins
- RMI over IIOP
- Security
- Servlets, JSPs and Web Applications
- System Administration
- Web Services
- WebLogic Tuxedo Connector
- XML

## Classloader

---

<b>Change Request Number</b>	<b>Description</b>
063742	There was a <code>java.lang.VerifyError</code> thrown during skeleton generation for the Remote interface of Entity bean. This has been fixed.
064273	In the Classloader, <code>weblogic.utils.classloaders.GenericClassLoader</code> , the <code>definePackage()</code> method now sets the package information according to the manifest.
069506	<code>ClassLoader.getResources</code> is now working when trying to access an XML file which is in the Class-Path: entry of the EJB's manifest file.
075553	Fixed a memory leak that was occurring because the core classloader was being stored unnecessarily.

---

## Cluster

---

<b>Change Request Number</b>	<b>Description</b>
072281	<code>HttpClusterServlet</code> now works properly when <code>WAPEnabled=true</code> .
072464	A <code>NullPointerException</code> no longer occurs during failover using SFSB handles created within the server.
073917	<code>NoSuchObjectException</code> was being thrown on failover from SFSB's handles in a cluster. This has been fixed.
073920	The Execute Queue length doesn't appear in the console and this functionality is not implemented with WebLogic Server 6.1.
074058	When using the IP name instead of the DNS name for the <code>WebLogicCluster</code> parameter, the secondary was marked NONE if the primary went down. This has been fixed.

---

---

074236

Added the attribute, `MemberWarmupTimeoutSeconds`, to `ClusterMBean`. The default value is 0 which results in no cluster warm-up. By default, the cluster start-up sequence remains unchanged. If a value  $> 0$  is set, the cluster members wait for that period of time to synchronize with other members during start-up. All `MessageDriven` Beans are deployed after the warm-up time. Only then do we start accepting client requests. There is no separate property for delayed `MessageDriven` Bean deployment.

---

## Console

<b>Change Request Number</b>	<b>Description</b>
051136	Broken monitoring link removed from the console.
055821	Forced the Public ID to be non-null for editing.
056161	Using the console EJB runtime monitoring customization, it is now possible to view the machine on which the EJB is deployed.
058271	Using the console to modify the deployment descriptor no longer changes the bean's manifest file.
059784	<code>wls_management_internal{1,2}</code> is no longer selectable as a <code>DefaultWebApp</code> .
059963	Added Max Open Sock Count to server->config screen in the console.
060116	The admin console no longer displays the Automatic Deployment checkbox when you startup WebLogic Server in production mode.
061664	The server shutdown message is now correctly displayed using a Netscape browser.
061686	Fixed broken deployment descriptor help links.
061838	It is now possible to start the managed server without restarting the admin server.
062020	The console XML editor is now aware of <code>&lt;validate-db-schema-with&gt;</code> .
062788	Fixed the Netscape default value for the refresh button so that a page in the security console was not re-appearing every time you clicked the OK button.
062991	Added information to the console help files to explain how to set up the five fields for using Node Manager to remotely start the managed server.
063543	The <code>ConsoleMainAttribute</code> for servlet monitoring in the console is now enabled.
064849	Modified the <code>weblogic.xml</code> descriptor editor to handle the <code>PersistentStoreTable</code> parameter.
067101 & 073625	Remote context viewing the local JNDI tree via the console is now possible.
067362	The console no longer references the wrong login timeout attribute.
070018	Start Default Console no longer fails when default browser is Netscape 6.2.

---

072909	The correct JMS pool name is now displayed in the runtime table.
073536	The console now has a check box that allows for optional mutual authentication.
074693	The link used to save realm implementation changes is now more explicit.
075338	The console now displays the correct information during monitoring of a Web Application.

---

## Core

---

<b>Change Request Number</b>	<b>Description</b>
CR073529	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-20.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-20.jsp</a> .

---

## EJB

---

<b>Change Request Number</b>	<b>Description</b>
047305	Fixed an error in the <code>TxCommits</code> value in Console.
054789	The message-driven bean no longer throws a <code>NoSuchElementException</code> and the message listener is no longer being removed from the list before the listener was finished processing it.
056081	Deployment of EAR stops short of setting component target.
057608	Now able to save more than 4K of data with CMP beans.
059045	<code>run-as</code> works in <code>ejbCreate()</code> .
059250	A <code>NullPointerException</code> no longer occurs in <code>ejbc</code> when the persistence element is missing in WLDD.
060720	<p>In 6.1 service pack 2, if you are using Blob/Clob, you have to set <code>&lt;isolation-level&gt;</code> to <code>TRANSACTION_READ_COMMITTED_FOR_UPDATE</code>. Otherwise, you will get the following exception:</p> <pre>java.io.IOException: ORA-22920: row containing the LOB value is not locked</pre> <p>If this happens, when using Blob/Clob, the container will lock all the tables within the transaction. In 6.1 service pack 3, you don't need to set <code>&lt;isolation-level&gt;</code> to <code>TRANSACTION_READ_COMMITTED_FOR_UPDATE</code> anymore. The container will only lock the table that contains the Blob/Clob, and the other table within the transaction will not be locked. This make Blob/Clob much easier to use.</p>

---

060738	Multiple phantom message-driven bean connections are no longer created without new corresponding sessions being added to the list after a JMS client reconnects to a restarted JMS server.
060867	EJB2.0 deployer no longer throws <code>AssertionError</code> when <code>RemoteMethod</code> name starts with 'find'.
061700	Some identifiers in generated SQL Query were missing quoted identifier escape which may have caused the generated EJB CMP SQL to fail to escape the Quoted Identifiers. This has been fixed.
062256	The <code>-c</code> flag in <code>DDConverter</code> now converts the input JAR files into <code>ejb20</code> and combines the JAR files. <code>DDConverter</code> would not work with the destination JAR file specified as follows: <code>java weblogic.ejb20.utils.DDConverter *****-c new.jar***** -d . convert.jar</code> .
062481	Added an attribute in <code>weblogic-ejb-jar.xml</code> to indicate if EJB is collocated with web tier.
062518	<code>DDConverter</code> no longer throws an exception when it encounters a <code>findAll</code> finder query.
062626	An empty <code>ejb-ql</code> tag is now allowed.
062676	Load-balancing of Stateless EJBs in a cluster no longer fails for random load algorithm. The deployment element <code>home-is-clusterable</code> can be defined for stateless session beans in <code>weblogic-ejb-jar.xml</code> .
062916	In 6.1 service pack 2, a caller's transaction was not being resumed after a transaction started on the server completed. When this happened, the caller's transaction would never complete. This problem would only occur when a remove method that had a transaction attribute that would cause the client's transaction to be suspended was called on an <code>EntityBean</code> in a transaction and a new transaction was started on the server to service the call. Transaction attributes that cause this behavior are <code>NotSupported</code> and <code>RequiresNew</code> . In 6.1 service pack 3, a caller's transaction is being resumed after a transaction started on the server completes.
062974	When a single-digit finder was defined on a 1.1 CMP bean, it did not throw a <code>FinderException</code> when more than one result was returned from the query. This has been fixed.
063077	The transaction time out attribute in <code>weblogic-ejb-jar.xml</code> now works.

## 6 Resolved Problems

---

063146	When a BMP EJB's concurrency strategy was set to Exclusive and the transaction attribute was set to NonSupported, a different instance was assigned when the transaction timeout threshold was met. This has been fixed. In addition, a remote exception is no longer being thrown when the local transaction is rolled back.
063387	WebLogic Server no longer fails to clear a lock on an entity bean after a deadlock situation. The waiters in the LockManager are now being cleared.
063469	A NullPointerException is no longer thrown while creating CMP EJB relations.
064293	ExclusiveLockManager is now removing waiters from the list even if the waiter happens to be TimedOut.
064425	Each ejbHomeMethod call was creating a new bean instance instead of utilizing pooled bean instances. This has been fixed.
064447	When decoding an EJB object handle from a cookie, an InvalidClassException is no longer being thrown.
064561	When <finders-load-bean>=false, new bean instances were being created disregarding pooled instances. This has been fixed.
064625	No longer getting 'local interface must not include java.rmi.RemoteException in its throws clause' from the JDBC when this should not be gotten.
064967	The ejbc no longer fails due to empty strings environment variables.
064969	The DDConverter now generates correct assembly descriptors.
065092	For EJB1.1 beans in 6.1 service pack 2, in a single transaction if the persistent field of a bean is changed and then a finder is invoked which returns the same bean, the new value of persistence field is overwritten by the value retrieved from the database. In 6.1 service pack 3, changes made to the EntityBean are no longer lost in the middle of a transaction.
065290	EJBC/RMIC generates XML descriptors with clusterable=true even for beans for which clusterable is false.
065321	Fixed a locking issue with an entity bean.
065571	When the entity bean didn't have a create method, DDinit was not generating the Descriptor files. This has been fixed.
066245	The ctx.getCallerPrincipal no longer returns run-as principal.

066644	A JMS message-driven bean and Queue with persistent file store was carrying over the transaction context in <code>onMessage</code> . This has been fixed.
067018	Calling <code>EntityContext.getEJBLocalHome</code> from the home method was throwing an <code>IllegalStateException</code> . This has been fixed.
067515	When invoking a JSP during 5.1/6.1 interoperability testing, the first time succeeded and the second time produced an error. This has been fixed.
068262	For EJB1.1 beans we were using <code>setBytes</code> to insert a value into a LONG column. This failed when the data was larger than 4K. We are using <code>setBinaryStream</code> now, which avoids causing the following exception: <code>ejbCreate(): java.sql.SQLException: ORA-01461.</code>
068569	A <code>tmp_ejbc</code> directory was being created even if not using server-side <code>ejbc</code> . This has been fixed.
068971	The <code>ejbc</code> no longer generates JDBC code for CMP EJBs with deprecated methods.
069280	An Illegal Reentrant Error is no longer incorrectly thrown by WebLogic Server during concurrent client calls to an entity bean.
069391	Fixed the CMP Manager <code>cascadeDeleteRemove</code> which was broken for CMP Partners.
070099	DDConverter fixed to generate the method-permission tags.
070565	Added a property to <code>weblogic-ejb-jar.xml</code> , called "clients-on-same-server", to indicate that the EJB is collocated with the web tier. This is a performance fix, designed to reduce multicast traffic. The default is false.
070754	In 6.1 service pack 2, <code>afterCompletion</code> was being invoked on a <code>StatefulSession</code> bean while the bean method was still busy processing. This behavior is counter to the EJB 2.0 specification. In 6.1 service pack 3, serializing Business methods and EJB Callbacks are being invoked which means that <code>afterCompletion</code> will not be invoked on a <code>StatefulSession</code> bean until the bean method has completed processing.
071027	Previously, the container checked for the CMP table by firing an SQL query ( <code>select * from tableName where 1=0</code> ). This way of checking is not efficient in the case of DB2 where the query generates a full table scan and is very slow if table has a lot of data. The container now provides an option to check the table using <code>DatabaseMetadata</code> . Although usually slower than checking using an SQL query, this way works well for DB2. See the <code>&lt;validate-db-schema-with&gt;</code> tag in the <code>weblogic-rdbms-jar.xml</code> file for more information.

## 6 *Resolved Problems*

---

---

071296	When doing an <code>ejbCreate()</code> on an entity bean ( EJB 2.0 ), a <code>java.lang.IllegalStateException</code> was being thrown. This has been fixed.
071377	Now WebLogic Server supports the ability to generate the EJB QL extension ? IS [NOT] NULL.
074710	An error no longer occurs while running a stateful session bean in a cluster. When a stateful session bean was passivated, the remote entry was removed and when the bean was activated, the remote entry was not returned, resulting in a <code>NullPointerException</code> .
075867	Message-driven beans deployed on WebLogic Sever had a problem re-establishing a connection to an MQSeries JMS server. Extra connections were created because of the number of triggers scheduled. This has been fixed.
076167	In 6.1 service pack 2, if an <code>SQLException</code> was thrown in <code>ejbLoad</code> of a read-only bean, the caller's (global) transaction was not being resumed. In 6.1 service pack 3, it is being resumed.
078147	Fixed a problem that was causing the following exception when using an EJB11/CMP11 bean:  Exception in <code>ejbCreate()</code> : <code>java.sql.SQLException: ORA-01461: can bind a LONG value only for insert into a LONG column</code>

---

## Examples

Change Request Number	Description
048185	The ACL example is now documented with the client example.
053467	The second compilation of the <code>i18n.logging.message</code> directory no longer gives errors.
056884	Fixed an error in the instructions for an example.
057904	Changing the connection pool is now mentioned in <code>petstore.html</code> .
062545	A WebLogic Server domain running Solaris <code>simpappcns</code> could not call a Tuxedo Corba Server domain running on NT. <code>simpappcns</code> now works for multi-platform environments, for instance, Tuxedo running on Solaris, and WebLogic Server running on Windows NT.
062962	The username/password authentication in the JAAS example now works on a UNIX environment using the <code>CertLoginModule</code> .
062999	Fixed a problem that was occurring when establishing a WLEC connection between WebLogic Server and Tuxedo. The WLEC <code>simpapp</code> example now connects to Tuxedo service.
063229	Made improvements on <code>examples/iiop/ejb/stateless/server/tux</code> .
071478	JMS Draw demo example is no longer broken with unique windows.
077055	Fixed a problem that was causing the Petstore server to begin with an error making it impossible to deploy Petstore.

## JDBC

Change Request Number	Description
052393	Updated the thin driver so that it does not have an error in JDBC session persistence while putting <code>StringBuffer</code> object.

## 6 Resolved Problems

---

054864	In 6.1 service pack 2 the server was hanging when using MultiPool with more than one database under heavy load. The MultiPool request would hang because the database was not responding which resulted in the whole server locking up at <code>weblogic.jdbc.common.internal.MultiPool.searchHighAvail(MultiPool.java:200)</code> . The cause was over-synchronization in the code which has been fixed in 6.1 service pack 3.
057340	When a connection pool is empty, the message now says "No available connections in pool."
057977	Connection pool does not rollback anymore on cleaning up JTS connections
059020	Deadlock no longer occurs if <code>weblogic.jdbc.common.pool.shutdownSoft()</code> is called when the connection pool is used by another user.
061786	<code>CREATE_POOL</code> now fires the connection pool just after <code>DESTROY_POOL</code> .
064198	Several classes were directly referencing optional Oracle classes in the server side code. This no longer occurs.
066001	Added a property to the <code>JDBCConnectionPool</code> <code>XAPassword</code> to encrypt the database's password.
066964	When using the WebLogic jDriver for Oracle/XA for a JDBC connection pool, refreshing the connection pool after a DBMS failure and restoration occasionally resulted in a core dump. This has been fixed.
066966	The <code>ResourceAllocator</code> monitor is no longer held while making a JDBC call to test table.
068490	Synchronized the call to <code>xa_open</code> at the java level.
068952	Provided automated build scripts for jDrivers.
070209	It is now possible to use the Oracle 9.0.1 Thin Driver to connect to an Oracle database. To use the driver, include the path to the driver class files ( <code>classes12.zip</code> ) to your <code>CLASSPATH</code> before the entry for <code>weblogic.jar</code> .
071974	<code>PreparedStatement</code> cache size set to 0 by default.
073640	Added a new attribute for a JDBC connection pool: Open String Password (or <code>XAPassword</code> in the <code>config.xml</code> file). This new attribute allows for encryption of the database password in the open string for connection pools that use an XA JDBC driver that requires an open string. For more information, see <a href="#">Database Passwords in Connection Pool Configuration</a> in the <i>Administration Guide</i> .

---

076090

Improved the behavior of database passwords included in the Properties attribute of a JDBC connection pool. If you specify a database password as part of the connection properties or open string, WebLogic Server parses the values and moves them to the Password and Open String Password attributes, respectively, which are encrypted when stored in the `config.xml` file. For more information, see [Database Passwords in Connection Pool Configuration](#) in the *Administration Guide*.

---

## jDriver

Change Request Number	Description
056531	There is no longer a performance regression in jDriver.
061643	<code>weblogic.Admin RESET_POOL</code> no longer fails after the database recovers.
063872	When a string value is returned from the out parameter of a stored procedure, trailing spaces are no longer stripped.
063874	<p>The time component of an SQL DATE was not removed when the JDBC <code>getDate()</code> methods were used. This behavior was non-compliant with the JDBC 2.0 specification for <code>java.sql.Date</code>: "To conform with the definition of SQL DATE, the millisecond values wrapped by a <code>java.sql.Date</code> instance must be 'normalized' by setting the hours, minutes, seconds, and milliseconds to zero in the particular time zone with which the instance is associated."</p> <p>A correction was made to the WebLogic jDriver for Oracle. Hours, minutes, seconds, and milliseconds are now removed from the returned value of the JDBC <code>getDate()</code> methods.</p>
069109	Modified JDBC test scripts to make Oracle 8.1.7 the default.
069676	The WebLogic jDriver for Oracle now supports the <code>we8iso8859p15</code> character set.
070878	If <code>PreparedStatement.setString</code> was used with an incorrect positional index of '0' with a DBMS connection created using the WebLogic jDriver for Oracle, the JVM would crash. Now, the jDriver gracefully handles the coding error and throws an exception.
078936	<p>In WebLogic Server 6.1 Service Pack 2, application of the HP-UX UNIX patch (PHSS_24627) caused an HP Server Bus error when WebLogic Server 6 accessed Oracle upon start-up. With the newer <code>aCC</code> libraries version A.03.33, using the <code>-AA CXXFlag</code>, the user gets a <code>java.lang.UnsatisfiedLinkError</code> while accessing Oracle upon WebLogic Server start-up.</p> <p>This problem was solved.</p>

# JMS

Change Request Number	Description
044976	Made sure that returned values and exceptions from any get method in <code>MapMessage</code> conform to the JMS specification.
058876	Disabling or enabling the JMS server through the Administration Console was causing WebLogic Server to throw an exception for any registered durable subscribers. This has been fixed.
061094	In a clustered environment, a <code>NoSuchObjectException</code> is no longer thrown for JMS clients listening on a JMS server if one of the WebLogic Server nodes in the cluster is not immediately restarted after shutting down.
061552	JDBC connection is no longer reset after the server is idle for three hours.
062669	JMS now supports selectors with unicode strings. Specifically, the expression parser now uses a <code>StringReader</code> and the <code>charVocabulary</code> range was increased.
062744	A NULL value is now accepted for a JMS <code>MapMessage</code> property to conform with the JMS specification.
064727	Fixed a problem with creating <code>ObjectMessage</code> for JMS in an applet.
067134	Added <code>weblogic/management/configuration/JMSConstants.STORE_ENABLED_XXX</code> .
067286	JMS paging no longer fails to page during a server restart.
068667	Upon booting, JMS non-persistent-message-paging file stores was failing to make available all disk-blocks for re-use. This has been fixed.
069757	JMS with SSL no longer throws the <code>weblogic.jms.common.LostServerException</code> .
070454	Message recovery no longer fails nor is delayed upon server crash when using message-driven beans with durable subscription.
072556	No more JMS Queue deadlock in <code>addReader()</code> .
073403	Fixed a problem that was due to a JNDI object having the same JNDI name of a different type.

## JTA

---

<b>Change Request Number</b>	<b>Description</b>
062681	WebLogic Server no longer fails on JDBC connection request to the pool whenever the connection object is left unclosed in the JDBC code.
063053	The XA connection can now specify the <code>openString</code> property.
064232	The <code>javax.transaction.HeuristicMixedException</code> is no longer being thrown sporadically at commit.
064403	After a Transaction timeout using the XA driver and a Stateless Session Bean, the XAConnection became unusable until WebLogic Server was restarted. This has been fixed.
064825	Method descriptors for <code>SubCoordinatorImpl</code> are now getting parsed at runtime.
065495	Added a system property to allow users to set the amount of time they are willing to wait to obtain the Coordinator reference. To specify the wait time, specify <code>-Dweblogic.JTA.ContactCoordinatorWaitSeconds=&lt;seconds&gt;</code> when starting the client or server.
066420	JTA server-resource cache was not being updated correctly on a member server <code>peerGone</code> event. This has been fixed.
068447	XA transactions with <code>TestConnectionsOnReserve="true"</code> no longer fails.
070855	JTA server-resource cache updates delayed under load no longer causes aborted transactions.
076439	Fixed a problem that was causing random XA transactions to be rolled back with an <code>Illegal State (Expected: preparing)</code> message.

---

## Miscellaneous

---

<b>Change Request Number</b>	<b>Description</b>
------------------------------	--------------------

---

042372	The API URL ( ) provided in WebLogic now functions properly. For some URLs, it was causing an exception error.
050388	Statistical information for the Jolt connection pool is now being displayed in the console.
052478	<p>A timing related JVM bug on Solaris is encountered that causes the VM to exit during object de-serialization of management objects at server startup. If you run into this JVM bug:</p> <ul style="list-style-type: none"><li>■ Re-start the server.</li><li>■ Upgrade the Virtual Machine to JDK version 1.3.1_04.</li></ul>
053029	The RA component is now removed during delete.
054080	In 6.1 service pack 2 on a Solaris machine, when WebLogic Server was using SSL to communicate (both as a client and a server) and was using <code>NativeIOEnabled="true"</code> , the JVM instance consumed more and more memory and eventually hit the heap size limit. This bug was encountered at a WebLogic Integration installation which relied on SSL communication between trading partners. In 6.1 service pack 3, this problem no longer exists.
055007	Created a configurable parameter, <code>weblogic.system.openSockCount</code> , that would determine the number of allowable connections.
056622	Added a new system property: <code>weblogic.net.http.URLStreamHandlerFactory</code>
056698	Fixed a problem with installing customer protocol handlers.
057313	Fixed <code>ConcurrentModificationException</code> while iterating through <code>AbbrevMap</code> .
057357	Fixed a problem that occurred when defining an error page 401.
057743 & 060981	The WebLogic Server <code>verboseToZip</code> and <code>logToZip</code> utilities have corrected problems with versioning information. Manifest entries, on which WebLogic classes depend for versioning information, are now pulled from <code>weblogic.jar</code> . The <code>verboseToZip</code> output file no longer includes entries for dynamic proxy classes. If a message log contains any instance of <code>i18n</code> messages, then <code>logToZip</code> now pulls all of the <code>i18n</code> related files from <code>weblogic.jar</code> to the client jar.
058327	Getting <code>ClassCastException</code> when trying to make a look-up to <code>DataSource</code> from an Applet
058358	The <code>&lt;alt-dd&gt;</code> tag in the <code>application.xml</code> file is now being recognized.

## 6 Resolved Problems

---

059054	Added the following symbol as a cookie separator: ",,"
059104	A property called <code>javaHome</code> which specifies the JVM that is used to start up a managed server was added for node manager.
060062	The Connections High column in the connection pool monitoring screen no longer includes connections refreshed by WebLogic Server.
060079	When using JMS over t3s, a <code>MaxMessageSizeExceededException</code> was being randomly thrown even though the message size never exceeded 50k.
060211	Fixed the placement and quotation of startup arguments passed to node manager from the Remote Start tab in the Console.
060739	<code>UserTransaction</code> now works properly under load. Now, transaction-managed EJBs control the isolation level automatically if the EJB starts the transaction as well as if the user starts the transaction.
061059	It is no longer possible to find a record after it has been removed from a database.
061458	WebLogic Server now supports starting <code>NodeManager</code> on Solaris and HP-UX without using Native libraries.
061512	Fixed problems starting WebLogic Server as a service on Windows NT and Windows 2000.
061847	There are no more error messages when using HTTP with <code>examples.io.FileBrowser</code> .
062086	Using <code>DDConverter</code> to convert from CMP 1.1 to CMP 2.0 no longer throws exceptions for JAR files that contain more than one EJB.
062177	There are no longer any peer to peer connections in clustered nodes.
062375	<code>PeerInfoable.getPeerInfo</code> no longer returns null when marshalling a transaction's propagation context in a clustered environment.
062439	Added error handling code in the native layer of Windows.
062565	The 1.2 JVM client no longer causes a socket connection leak.
062752	<code>jDriver</code> was giving an <code>ORA-24347</code> error on an outer join operation which selected more than 100 rows. This has been fixed.
062926	JSP page tags can now handle MS932 specific characters on Solaris or HP.

062989	A JMS client running on a slow modem connection and trying to establish a JMS topicConnection with a cluster of two WebLogic Servers no longer gets a ClassCastException.
062997	The servlet engine now removes the servlet stub of the generated JSP file if the source file has been deleted.
063103	Context lookup was failing within a spawned thread. This has been fixed.
063147	Applets no longer fail during downloading of Home interface Stub with the JDK 1.3.1_01 plug-in.
063310	CGIServlet no longer fails under load due to File Descriptor leak.
063354	No longer receiving a ClassCastException from rjvm.MsgAbbrevInputStream.readClassDescriptor() when looking up a session bean from a servlet.
063671	Client side using HTTPS tunneling was producing the get java.net.ProtocolException: Tunneling result not OK, result: 'DEAD'
063830	No longer throwing a javax.transaction.TransactionRolledbackException when the cache is full.
064117	Fixed a problem that was causing the server to abort the connection and result in a java.net.SocketException.
064125	Please review the security advisory information at <a href="http://dev2dev.bea.com/resource/library/advisories/notifications/BEA02-15.jsp">http://dev2dev.bea.com/resource/library/advisories/notifications/BEA02-15.jsp</a> .
064130	ReplicaLists are now serializable in order to allow more than one lookup under IIOP.
064391	When deploying applications that contain malformed MANIFEST.MF entries, exceptions are no longer being thrown.
064404	When the same instance of IE is used to configure a webApp using the console and then to access the same webApp (containing French characters for instance), sometimes accented characters were not being displayed correctly. This has been fixed.
064434	If a servlet used RequestDispatcher.include() to include the output of a JSP, the post-processing filter was not able to add a header to the response. This has been fixed.

## 6 Resolved Problems

---

064860	WebLogic Server no longer fails to execute CGI scripts with JSPs that use page buffering.
065697	When SetUID was enabled, the server wouldn't start. This has been fixed.
066130	Corrected an error message that was being thrown by the ejbc during compliance checking.
066158	When a Log Filter is defined and targeted to the managed server, the trap is now invoked when using the NonCatalogLogger.
066252	Using a specific Factory/Trader pattern, a ReplicaAwareRemoteRef warning log was being displayed on the client side. This log is no longer displayed.
066504	Updated the SP installer with JDK 131_02 for Solaris and Windows.
067508	A generic classloading exception was being thrown when generating a thin client with AppletArchiver. This has been fixed.
067516	The appletArchiver now spools the i18n and manifest files to the thin client jar.
067517	AppletArchiver no longer generates L10n type exceptions when building the client jar.
067648	Fixed a problem that was causing WebLogic Server to hang due to the PosixSocketMuxer code.
068570	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-16.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-16.jsp</a> .
069043	When using HttpProxyServlet in conjunction with a filter, a SocketResetException no longer occurs in NTSocketMuxer.
069675	Reading from a file with @file option on the BEASVC command line no longer has a limitation of 4K.
069991	No longer receiving a ClassCastException from <code>rjvm.MsgAbbrevInputStream.readClassDescriptor()</code> when looking up a session bean from a servlet.
070575	After the server had been running about two days, a client running on a separate JVM was not able to look up a bean due to the JNDI tree and JNDI objects eventually becoming out of sync. This had been fixed.
071580	WebLogic Time service thread is now setting Context Class Loader.

071822	In order for jCOM to successfully find the WebLogic license file, the beahomelist file must be located in C:\bea.
072010	Configuring WLEC connection pool was interfering with command line properties. This has been fixed.
072061	The node manager -Dweblogic.nodemanager.sslHostNameVerificationEnabled=true option now performs host name verification.
072211	Improved the Solaris native implementation of node manager for handling error conditions.
072228	WebLogic Server was not firing a trap for the JDBCConnectionPoolRuntime on the admin server. This has been fixed.
072278	Improved the HPUX native implementation of node manager for handling error conditions.
072327	Attributes with type LONG can now be monitored with SNMP.
072495	Fixed a memory leak caused by Timer Services.
072904	When CallRouter was used, ReplicaList was being corrupted. This has been fixed.
073116	Fixed the weblogic.Admin SERVERLOG command-line utility.
073201	When running multiple HTTP tunneling clients over a modem connection, the server was eventually hanging. This has been fixed.
073400	Modified verboseToZip to exclude javax classes from the output file to reduce the size of the client jar.
073529	Fixed a problem that was causing clustered servers to lock under heavy stress testing.
073569	Eliminated <DGCserver> info messages in the weblogic.log file in order to save space in the file.
073578	Installing Windows NT service by specifying JRE directory as its JAVA_HOME failed with an error when trying to start the service. This has been fixed.
073788	The HP Installer was failing to build. This has been fixed.
074991	Fixed a problem that was resulting in WebLogic Server crashing in the performance pack under heavy load.

## 6 Resolved Problems

---

---

075271	A <code>NullPointerException</code> is no longer thrown at <code>weblogic.utils.BubblingAbbrevier.getValue</code> ( <code>BubblingAbbrevier.java:163</code> ). The Abbrev tree is no longer getting corrupted.
075700	For WebLogic Server instances that run as a Windows service, you no longer need Windows Administrator privileges to start the service.
076395	For WebLogic Server instances that run as a Windows service, using the Administration Console or the <code>weblogic.Admin</code> utility to shut down the server no longer causes the Windows service to report an error condition.
076705	Deploying a <code>MessageDriven Bean</code> no longer fails with a <code>NoAccessRuntimeException</code> for guest if the destination is remote.
077238	Startup scripts now specify the correct value for the Oracle <code>JDriver</code> path. Before this service pack, startup scripts specified <code>LD_LIBRARY_PATH=oci816_8</code> , which is no longer supported. Now the scripts specify <code>oci817_8</code> .
077278	Removed the Oracle 816/806 driver from the WebLogic Server 6.1 service pack 3 installation.
077919	Fixed a problem with an earlier bug fix that was causing calls to <code>ObjectOutputStream.writeObject()</code> on <code>EJBHandle</code> to throw the following error: <code>Closing: 'weblogic.rjvm.t3.T3JVMConnection@49c89f'</code> because of: <code>'Server received a message over an uninitialized connection.</code>
078523	The java version bundled with the WebLogic Server 6.1 service pack 3 installer is: <ul style="list-style-type: none"><li>■ Win 2k &amp; Solaris installer bundled with java version 1.3.1-03</li><li>■ HP installer bundled with java version 1.3.1.05.</li></ul>

---

## Plug-Ins

Change Request Number	Description
054344	WebLogic Server can now proxy all requests except a certain MIME type.
057448 & 059142	Fixed a memory problem that occurred when using SSL under heavy loads.
058886	Fixed an IIS Proxy memory leak with SSL.
060325	Concurrent GET and POST requests led to a hang in the Apache plug-in
060863	Fixed a problem with the NSAPI and ISAPI plug-ins; a POST timed-out error was being written to the browser even after a second retry worked and this HTML content was printed in the browser.
061229	A request is now declined if handler is not set to <code>weblogic_handler</code> explicitly.
061379	<code>RequireSSLHostMatch=true</code> now works with NSAPI.
061386	<p>A new configuration attribute, <code>WeblogicPluginEnabled</code>, has been added. This attribute can be configured at the server or cluster level.</p> <p>When this attribute is true for a server (or cluster) that receives requests from a proxy plug-in (or <code>HttpClusterServlet</code>), a call to <code>getRemoteAddr</code> will return the address of the browser client from the proprietary <code>WL-Proxy-Client-IP</code> header, instead of the web server.</p> <p>For non-clustered servers that will receive proxied requests, this attribute may be set at the server level, on the <code>Server --&gt; Configuration --&gt; General</code> tab.</p> <p><code>WeblogicPluginEnabled</code> is duplicated in <code>ClusterMBean</code> and <code>ServerMBean</code>. <code>ClusterMBean</code> overrides <code>ServerMBean</code>. The default value is false.</p>
061561	The request is not timed-out in case of failover from a hung server.
061881, 065980 & 076503	The NSAPI Proxy plug-in no longer fails to handle large (greater than 2k) POST Requests.
062607	When the <code>KeepAlive</code> is enabled on the plug-in IIS, the connection is now closed immediately with a HEAD http request.
062641	With the IPlanet plug-in, from a Java client using JSSE, the servlet access now works.
062778	Plug-in is now able to proxy requests to the primary/secondary servers without checking if those are in the current server list.

## 6 Resolved Problems

---

062808	Added support for the latest beta version of Apache 2.0.
063004	The Apache proxy plug-in no longer generates an incorrect URL under a heavy load.
064268	In the response header the status, HTTP/1.1 200 OK, the string "OK" is no longer being stripped off by the <code>iisproxy.dll</code> .
064472	Apache 1.3.12 EAPI, Solaris 2.8, with <code>mod_wl_ssl.so</code> was generating "Segmentation Fault - core dumped" while using port based virtual host. This has been fixed.
064890	Cookie Path no longer gets mangled when <code>PathTrim</code> is used in Apache.
065188	Fixed the iPlanet plug-in so that it no longer gives a <code>WRITE_ERROR</code> with a 404 error for a static page.
065443	Now able to proxy to both 5.1 and 6.1 back-ends with the NSAPI plug-in.
065660	The HTTP header is malformed with the last ISAPI plug-in.
067100	The build information of a plug-in now includes the version number.
068243	ISAPI plug-in <code>PathPrepend</code> path is no longer prepended twice when using FORM authentication for a web application.
068517	The proxy plug-ins no longer load the ECC crypto module.
068562	When setting up IIS to proxy to multiple sites, the plug-in was not working when putting a firewall or a load balancer in front of the IIS server. This has been fixed.
068634	A 413 error is now returned if Content Length exceeded the plug-in parameter <code>MaxPostSize</code> .
068790	Now supporting Apache2.0.32.
068985	Added the ability to exclude certain requests from being proxied in WebLogic Server.
070358	Apache-plug-in no longer gives response code 500 when the server is not available.
072895	When secondary is killed, request was not going to the same server (primary). Now it is.
073838	When using iPlanet, WebLogic Server no longer loses the HTTP session ID if <code>CookiesEnabled</code> is set to false.
073936	Apache Plug-ins have been ported to OpenVMS 7.3 platform.
074137	The NSAPI plug-in was returning "unknown" if <code>x-forwarded-for</code> and <code>client-proxy-ip</code> headers were not sent or set. This has been fixed.

074286	Now able to proxy to both 5.1 and 6.1 back-ends with the Apache plug-in.
074393	Added URL parameters to the correct <code>requesturi</code> variable because the latest iPlanet plug-in was dropping URL parameters.
074656	If the password to start WebLogic Server was provided on the command line, the property, <code>weblogic.security.jaas.Configuration</code> , was not initialized at all. This has been fixed.
076457	Fixed the problem with <code>READ_TIMEOUT(os error=22)</code> on NSAPI for Solaris. The NSAPI plug-in is now stable under a heavy load (800 or more users).
076578	The NSAPI plug-in no longer keeps trying to connect to a dead server.
076701	The Apache server no longer crashes when Content-Length is zero in a POST request by requiring that the server not read post data if Content-Length is 0.
076722	The Apache server no longer crashes when Content-Length is zero in a POST request.
076731	The NSAPI plug-in can now fail over to the active server when the primary and secondary servers are locked if it is specifically requested to do so.
076996	Fixed an SSL memory leak problem.
077440	The IP address of the proxy is now in the HTTP request headers.
077794	The keep-alive feature for Apache 2.0 is disabled for this release.
078265	<code>MaxSkips</code> is used if <code>DynamicServerList=OFF</code> .
078883	The NSAPI plug-in no longer dumps core when URL rewriting is used with a long URL.

## RMI over IIOP

<b>Change Request Number</b>	<b>Description</b>
047057	Support for complex objects is fixed in IIOP implementation.
062082	Improved parsing of IIOP URLs.
067013	Now, only if debugging is turned on, are <code>onMessages</code> printed whenever there is failure to renew lease.
072179	Resolved the difference in behavior for JDBC connections taken from pool driver or <code>DataSource</code> when not closed.
073028	Now, chunk continuations before encapsulation body are allowed.
073889	If <code>MuxableSocketIIOP</code> received more than one message in a chunk and if the overshoot was less than the length of the message header, then subsequent attempts to read the message length were wrong. This has been fixed.
074100	Muxer boundary condition no longer causes garbled message.

## Security

<b>Change Request Number</b>	<b>Description</b>
061659	WebLogic Server now starts with <code>NTrealms</code> configured
061694	When running the <code>SSLClient</code> example, a <code>NullPointerException</code> was being thrown at <code>weblogic.security.provider.MD2.&lt;init&gt;(MD2.java:24)</code> . This has been fixed.
062172	Now, if cookies are disabled in a browser, form-based authentication redirects to the secure JSP/Servlet.
062261	Fixed a security issue with EJB <code>SessionContext.getCallerPrincipal()</code> .
062988	Fixed a problem with the NT realms and multiple PDCs.

063349	When implementing an <code>AuthFilter</code> class, the methods <code>doFailAuth()</code> and <code>doSuccessAuth()</code> are now being called.
063763	Two-way SSL java client connection now works with encrypted private keys.
064250	The web server is no longer vulnerable to session ID forcing. This applies to web applications protected with form-based authentication.
064263	Lockout Duration above 322122 will now lock out the user.
064285	When running the <code>SSLClient</code> with <code>ssl.proxyHost/ssl.proxyPort</code> , the <code>SSLHostnameVerifier</code> now works as expected.
064342	It is now possible to redirect the <code>echo \$password</code> command output to the Java interpreter to start WebLogic Server other than by using <code>password.ini</code> or by specifying the <code>-Dweblogic.management.password=\$password</code> property.
065046	<code>ManageableRealm.newAcl()</code> no longer returns NULL when the System password is changed.
065314	LDAP version 2 no longer causes issues with user names containing natural language.
066232	The 2-way SSL with RMI/IIOP <code>PingClient</code> example is now working.
066264	Fixed a problem with the lockout mechanism using Japanese Windows 2000. A 500 error was occurring.
066491	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-17.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-17.jsp</a> .
067726	Fixed a problem with form-based authentication.
068524	Two-way SSL with RMI/IIOP and EJB access and authentication is now working for the <code>SimpleCertAuthenticator</code> example.
069083	Fixed a problem with perimeter authentication.
069160	There is now a way to set <code>ClientCertProxyEnabled</code> for unclustered servers.
070870	<code>LDAPRealmv2</code> is now fully implemented.
071167	There is no longer a <code>ProtocolException</code> when tunneling HTTPS through iPlanet proxy server.
072096	Now able to start a managed server using SSL and <code>DebugRC4="true"</code> .
072916	Changed class <code>weblogic.security.X500Name</code> to implement <code>Serializable</code> .

## **6** *Resolved Problems*

---

---

075109	LDAPRealmv2 now caches memberships.
--------	-------------------------------------

---

## Servlets, JSPs and Web Applications

Change Request Number	Description
044528	Under certain situations, <code>isUserInRole</code> was returning false when it should have been returning true. This has been fixed.
051065	XML fragments on the XML view of a JSP page were not being passed to out. Non-standard fragments, which are neither standard nor custom tags (tag libraries), are now being sent to the current value of out as is.
054694	<code>RequestDispatcher</code> will now invoke filters upon include/forward properly.
055333	When detecting <code>contentType</code> directive at the middle of the JSP, the JSP compiler was failing.
056153	The File Servlet now supports initialization parameters as well as context parameters.
058352	It is now possible to access a web application that has a one-character name like "i.war" or "j.war".
059026	In a web application that contains a servlet, when the mapping defined for this servlet is in the form of <code>*.aBc</code> , accessing the servlet no longer fails.
059102	The compression filter now works correctly with <code>RequestDispatcher</code> . The forward was causing an invocation of the filter, but the servlet engine was sending the empty response back to the client. This has been fixed.
059219	WebLogic Server no longer returns the default port number if Host Header is missing.
060023	The <code>FileServlet</code> now includes the <code>SERVLET_PATH</code> when calculating the source filename which means that it is possible to only explicitly serve files from specific directories by mapping the <code>FileServlet</code> to <code>"/dir/*"</code> etc.
060184	<code>PathInfo</code> is now decoded and preserves characters in the original URI.
060351	The method <code>PageContext.include(String relativeUrlPath)</code> was not throwing any exception when the URL to be included was not found. Added a log message that appears in the context log file and indicates the parent resource and which resource could not be included. The log message appears in the <code>mycontext.log</code> file - where <code>mycontext</code> is the context-path of the webapp.
060825	<code>BodyTagSupport</code> method, <code>getBodyContent()</code> , incorrectly returned null when body tag was empty.

## 6 Resolved Problems

---

061377	jspc no longer generates EVAL_BODY_TAG instead of EVAL_BODY_AGAIN.
061782	<code>ServletResponseImpl.setContentType()</code> can now handle multipart type properly.
061784	While getting the log from a managed server, the admin server java process was pegged at 90%. This has been fixed.
061864	WebLogic Server now prevents the display of secure cookies.
061948	Added a new semantic-predicate to distinguish between tags with body-content set to "tagdependent" or "jsp". Added a new rule before the existing opening-tag rule in the lexer. When the tagdependent rule is invoked, it does the same as the ordinary open-tag rule, but also looks ahead and matches the content of the JSP until it finds a matching closing tag. Everything it finds is escaped for \n, \r, and \ and printed to "out". It then returns to the parent lexer rule which should immediately find the closing tag.
061968	HEAD method no longer returns request body with use of <code>jsp:include</code> .
062109	Fixed a problem with JSP scripting variable synchronization
062160	The parameters <code>&lt;compileFlags&gt;</code> , <code>&lt;encoding&gt;</code> , and <code>&lt;compilerSupportsEncoding&gt;</code> are no longer ignored during precompilation.
062289	WebLogic Server now decodes the spaces in a URL.
062395	<code>load-on-startup</code> in <code>web.xml</code> is now be optional
062536	The JSP response that contains Japanese characters is no longer garbled when filtered with <code>HttpServletResponseWrapper</code> .
062538	Pages will now be properly parsed regardless of the location of the <code>contentType</code> .
062542	Now, requesting <code>aux.jsp</code> / <code>AUX.jsp</code> / <code>PRN.jsp</code> / <code>prn.jsp</code> on a WebLogic Server that is running on a Windows box either succeeds or fails instead of staying in an undetermined state.
062579	CGI scripts will now be resolved properly even if <code>CGIServlet</code> is registered as a default servlet.
062896 & 063009	When the JSP parameter, <code>precompile</code> , was set to <code>true</code> in the web app (WAR) <code>weblogic.xml</code> , only the document root level JSPs were being recompiled each time the server restarted. The subdirectory JSPs were not being recompiled. JSPs are now recompiled, even during pre-compilations, only if they have changed.

062920	When specifying an <code>&lt;exception-type&gt;</code> in <code>web.xml</code> in a webapp (the exception class is located under <code>WEB-INF/classes</code> ) when the webapp is deployed, WebLogic Server was not finding the exception class and throwing a <code>ClassNotFoundException</code> . This has been fixed.
063007	Session table name for JDBC persistence is now configurable.
063127	<code>ServletInputStream</code> now returns data if the session is new.
063169	Having accented characters and a <code>Charset=UTF-8</code> no longer yields the <code>weblogic.utils.ParsingException</code> and the nested <code>TokenStreamException</code> .
063288	WebLogic Server now only looks for <code>SessionCookie</code> in the cookies if <code>CookiesEnabled</code> is true.
063414	When trying to migrate applications from WebLogic Server 5.1 to 6.1 SP1, there is no longer high CPU utilization.
063524	A <code>NullPointerException</code> is no longer thrown using <code>FileServlet</code> with <code>nativeIO</code> turned on.
063563	<code>weblogic.jspc</code> was not passing <code>-encoding JAVAENCODING</code> value to <code>javac</code> from the page tag. This has been fixed.
063904	The inclusion of each page directive in a JSP was leading to the insertion of an extra line feed character in the generated HTML output. If the HTML consisted of an image, the image lost its formatting. This has been fixed.
063925	If a language-specific character encoding is specified for a web application, binary data written to the response output stream is no longer garbled.
064294	Fixed a problem with the WebLogic Server <code>SessionID</code> reuse using a virtual host.
064383	HEAD request from an applet no longer issues a <code>ClassCastException</code> from <code>ClasspathServlet</code> .
064446	When using two levels of proxying, a call to <code>getRemoteUser</code> was always returning <code>NULL</code> . This has been fixed.
064449	Web application deployment was failing and resulting in a <code>StringIndexOutOfBoundsException</code> . This has been fixed.
064650	WebLogic Server threw a <code>NoClassDefFoundError</code> and was unable to load a class from a JAR file with a period in its name. WebLogic Server is now able to load a class from a JAR file with a period in its name.

## 6 Resolved Problems

---

064880	In form-based authentication, when the user directly went to the login page, WebLogic Server was forwarding the user to the welcome page. This has been fixed.
064988	JSP precompilation no longer aborts after a compile error when the -k flag is set.
065104	Fixed by 064650 and 065213.
065194	A multi-part request in the SSL process was failing through a slow connection. This has been fixed.
065213	A <code>NoClassDefFoundError</code> occurred if more than one jar file was defined in the manifest classpath entry. For example, when <code>xalan.jar</code> was in <code>WEB-INF/lib</code> of war file, this problem happened because it specified two jar files in the manifest classpath entry. WebLogic Server no longer throws a <code>NoClassDefFoundError</code> under these circumstances.
065924	In a non-clustered environment, web applications with cluster-specific properties can now be deployed using the new option: <code>replicated_if_clustered</code>
066395	<code>response.addHeader</code> was failing with null character being inserted erroneously. This has been fixed.
066500	Forwarded JSP pages will now terminate the execution of the current page.
066569	<code>request.getParameter()</code> is now returning the parameter for post request with Javaclient.
066708	<code>request.setCharacterEncoding</code> was not working for JSP includes. This has been fixed.
067001	When using the Http Proxy/Cluster servlet, a request was sent to some back-end server and the response body was not returned to the client if the request came back with a status of 100. This has been fixed.
067072	When a JSP had <code>pageEncoding</code> without <code>contentType</code> charset specification, the generated .java file was in VM default encoding and the compile failed. This has been fixed.
067073	Generated .java and -encoding for javac did not match when both <code>contentType</code> and <code>pageEncoding</code> existed. This has been fixed.
067077	When there was an encoding setting in <code>jsp-param</code> of <code>weblogic.xml</code> at a JSP file with <code>pageEncoding</code> , <code>pageEncoding</code> was not used. This has been fixed.
067292	Latin one characters in request parameters, when re-directing from one JSP to another, were getting lost. This has been fixed.

---

067505	Fixed a problem with port in Location header for an HTTP re-direct.
067748	It is now possible to compress the output stream of a servlet.
067948	Page buffer directive is no longer ignored if it is in an included JSP page.
068024	Removed the need to use sync blocks in <code>HttpServer</code> or <code>HttpClusterServlet</code> . A new enumeration is constructed each time it is required that references a Set that is guaranteed not to change.
068560	Certain JSP pages' source code was being shown on the browser. This has been fixed.
068648	Limiting maximum POST size now works properly.
068674	There is no longer a <code>ClassCastException</code> thrown when using <code>HttpProxyServlet</code> in conjunction with a filter.
068809	Now, the write signature is honored when using encoding that is different from the default.
068821	<code>getRequestURI</code> from Request Object is now returning the full length of the URI.
069304	<code>HttpClusterServlet</code> now re-reads the post data from the input stream up to the declared content-length.
069511	Error page attributes are now being set on request for JSP error directives.
069630	If the property of a custom tag in a JSP contained an escape sequence ( <code>\</code> ), the JSP was not compiling. This has been fixed.
069652	The response header ( <code>ContentType</code> ) and the <code>JSPWriter</code> now have the same character set while serving a page to client.
069809	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-03.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-03.jsp</a> .
069956	For replicated sessions, the open session count is incremented only once.
070090	When sending session cookies in the response to the client, WebLogic Server did not specify the cache-control appropriately to prevent caching proxies from caching the cookie. This has been fixed.
070132	It is now possible to request a filename with Chinese characters.
070151	<code>weblogic.jspc</code> no longer silently exits when <code>&lt;load-on-startup&gt;</code> contains an empty value.

---

## 6 Resolved Problems

---

070470	Subsequent reload/refresh was at times executed by the old code that was still deployed in the container. This is now prevented by reloading of the stale class in the case where the JSP fails to parse.
070755	No longer sending wlsproxy specific headers if the requests are not coming via wlsproxies.
070823	Eliminated inconsistent behavior of index directories parameter across WAR and EAR components.
071076	Fixed classloading behavior problem that resulted in having both child1 and child2 using the same classloader, BUT additional classloaders being created and destroyed.
071082	Fixed broken URLEncoding.
071634	Fixed chunked transfer encoding issues that occurred when WebLogic Server 6.1 was communicating with WebLogic Server 5.1.
072557	<code>response.addCookie(name)</code> now honors the order in which cookies are sent to the browser.
073203	Now able to use the + character in a <code>&lt;url-pattern&gt;</code> .
073380	Improved the error messages for <code>CGIServlet</code> .
073516	Fixed a problem with the <code>CGI NonParsedHeader</code> .
073792	Active sessions will no longer be invalidated.
073797	JSP <code>useBean</code> tag is now parsed properly.
074015	There was a <code>NegativeArraySizeException</code> if <code>PROTOCOL</code> is missing in the first line of the request. This has been fixed.
075201	The security-constraint is now working when the <code>url-pattern</code> refers to a two-character directory.
075607	Socket exceptions squashed by <code>PrintWriter</code> could have tied up execute threads in servlets and JSPs. This has been fixed.
075628	When a program submitted data to a servlet using <code>Content-Type</code> that was set to include a charset, <code>getParameterValues("...")</code> , returned every submitted value twice. This has been fixed.
076056	When a JAR file's <code>MANIFEST.MF</code> contained a JAR file that was in a different directory, WebLogic Server was throwing a <code>StringIndexOutOfBoundsException</code> and the web app was not being deployed. This has been fixed.

---

076476	When using multiple frames, the last <code>accesstime</code> set by one frame was not being honored by the other frame. This has been fixed.
076742	Added a check for the <code>&lt;env-entry-value&gt;</code> element.
077329	<code>HttpSessionListener</code> failed to work correctly when the session is set to replicated. This has been fixed.
078930	Requesting <code>http://host:port/webApp/aux%20.jsp</code> no longer hangs the thread servicing the request.

---

## System Administration

<b>Change Request Number</b>	<b>Description</b>
050136	You can now unset the property values for MBeans from the command line.
052774	Applications in the <code>config.xml</code> file that do not actually exist are no longer being deployed.
054836	When migrating a cluster of servers from WebLogic Server 4.5 or 5.1 to WebLogic Server 6.0 or 6.1, an <code>InstanceNotFoundException</code> was thrown. This has been fixed.
056158	Undeploying an enterprise application that contains a default web app caused an exception at undeployment. This has been fixed.
056832	The WLEC Connection Pool was not trying to start when in a cluster. This has been fixed.
058946	Now, JSPs that are pre-compiled on the admin server do not compile again on the managed server.
061426	Servlets load-on-startup after, not before, the Start Up classes have run
061676	The converter utility now puts all log files in the <code>SERVER/DOMAIN/logs</code> directory.
061703	<code>weblogic.deploy</code> was always using the default <code>.wlstaging</code> directory, even if the application was originally deployed using a path other than <code>.wlstaging</code> (i.e., the <code>config.xml</code> was hand-edited). This resulted in update not working for applications that were installed any other way than by using <code>weblogic.deploy</code> . This has been fixed.
062135	Re-deploying an exploded webapp that has a subdirectory with the same name as the webapp results in an error. This has been fixed.
062151	Now, managed servers on a remote machine retain webapp name/timestamps.
062749	It is now possible to update the webapp if the webapp is deleted, deployed, updated, and then updated again.
064101	The admin server JNDI now shows the managed servers' MBeanHome.
064280	Fixed a performance regression that occurred when using the <code>weblogic.refresh</code> utility to update JSPs on the managed servers.

---

064575	<code>weblogic.refresh</code> did not refresh JSPs and static files from Windows 2000 to a server running on Solaris. This has been fixed.
064722	It is now possible to deploy an EJB using <code>weblogic.deploy</code> if a deployed component does not exist at start-up time.
066480	Registering a custom MBean with an object name of null, the server was throwing an exception. This has been fixed.
066704	The <code>FailuresFatal</code> property was not getting automatically generated upon migration from WebLogic Server 5.x to WebLogic Server 6.x. This has been fixed.
067917	The property, Console Context Path, on the <domain>->Configuration->General page of the console application now enables a domain to be accessible by a contextpath other than console.
068231	Re-deploying an enterprise application to a managed server (using <code>weblogic.deploy</code> ) after modifying the context-root now makes the new context-root effective until a server restart.
068579	Redeployment to Virtual Host was returning <code>javax.management.InstanceNotFoundException</code> . This has been fixed.
068745	Increased the console speed in a 100-node cluster.
069145	For auto deploy in the applications directory, the admin server has been added to the targets if it's not empty.
069407	Multiple senders can now be saved in the <code>MBeanServerNotificationListener</code> 's list.
069686	It is now possible, using MBeans, to change the <code>weblogic.xml</code> cookie name after the web application is deployed.
071633	Fixed <code>svuid</code> for MBean interfaces.
072676	Added <code>LoadBeforeAppDeployments</code> to console for startup class.
072964	Fixed a problem that was causing a <code>javax.management.AttributeNotFoundException</code> when using the EJB Descriptor editor in the console.
073674	After <code>webapp</code> deployment, the <code>weblogic.xml</code> session persistence type is now changed.

---

## 6 *Resolved Problems*

---

---

075139	Fixed a problem that was preventing <code>weblogic.deploy</code> from deploying an application to a cluster.
075667	Managed servers were not discovered by the admin server when the admin server was restarted. This has been fixed.
077750	Fixed a problem with deploying an EJB when the JNDI name was already in use.

---

## Web Services

Change Request Number	Description
046215	<p>You can now correctly use the protocol attribute of the wsgen Ant task to specify that you want to use HTTPS (rather than the default HTTP) when invoking a WebLogic Web service.</p> <p>You must, however, perform the following additional steps if you want to use HTTPS:</p> <ul style="list-style-type: none"> <li>■ Add the following line of Java code to the client application right before creating a URL: <pre>System.setProperty( "java.protocol.handler.pkgs" , "weblogic.net" );</pre> </li> <li>■ If you are running the examples server with the SSL certificate shipped with WebLogic Server, you need to specify the following on the command line when running your client application: <pre>-Dweblogic.security.SSL.ignoreHostnameVerification=true</pre> </li> </ul>
062976	MSSoapClient.ConnectorProperty for AuthUser and AuthPassword is now passed to InitialContext of EJB.
065205	Web Services FaultHandler was never being invoked. This has been fixed.
065669	Now able to use security ACL on a WSDL URL.
066226	<p>WebLogic Web services now support two-way SSL authentication between client applications and RPC-style Web services.</p> <p>To use two-way SSL authentication, add the following Java code to your client application before you obtain the context you are using the look up your Web service:</p> <pre>InputStream certs[] = new InputStream[3]; certs[0]=new PEMInputStream(new     FileInputStream( "sample_key.pem" )); certs[1]=new PEMInputStream(new     FileInputStream( "sample_cert.pem" )); certs[2]=new PEMInputStream(new     FileInputStream( "sample_ca.pem" )); h.put( SoapContext.SSL_CLIENT_CERTIFICATE, certs );</pre>
066703	There were excessive error messages being generated on SOAP client. This has been fixed.

## 6 *Resolved Problems*

---

---

072788	Now, wsdl is generated with element instead of attribute for complex types.
--------	---

---

073897	SOAP Fault now uses CDATA.
--------	----------------------------

---

## WebLogic Tuxedo Connector

Change Request Number	Description
052022	Codeset translation between WebLogic Tuxedo domains and Tuxedo domains is provided using the WebLogic Server <code>weblogic.wtc.encoding</code> property. For more information, see <a href="#">Configuring WebLogic Tuxedo Connector for Non-ASCII Codesets</a> .
055170	Fixed problem with <code>Fchg</code> method of the <code>TypedFML(32)</code> class in the WTC package.
056230	<code>tbridge</code> no longer fails to initialize the <code>wlsErrorDestination</code> JMS queue
062843	Fixed a problem with the <code>Fdel</code> method of the <code>TypedFML(32)</code> class in the WTC package.
063290	When loading large code tables from Tuxedo to WebLogic Server, the duration of service calls ( <code>tpcall</code> ) is no longer dramatically longer with WTC compared to with Jolt.
066489	Now able to call an EJB from a Tuxedo client when using FML32 buffer.

## XML

Change Request Number	Description
061660	Using JAXP API to parse XML document, there is no longer the error message "Failed to open XML document".
062297	You can now set the cache memory size, cache disk size, and cache timeout interval for the external entity cache using the Administration Console. In previous WebLogic Server versions, these fields did not accept any values.
063433	Fixed a regression with the XSLT processor embedded within WebLogic Server 6.0. When an MFL file contained a group containing a String and either a Literal or XSLT processor embedded within a WebLogic Server 6.0 Filler field, the transformation was hanging.

## 6 *Resolved Problems*

---

067966	Due to a fix in the WebLogic Server classloader, you can now plug-in any version of the Xerces XML parser.
065345	Fixed a problem that was causing <code>CharReader.fillCurrentChunk</code> to throw a <code>java.lang.ArrayIndexOutOfBoundsException</code> .
067022	The transformer shipped with WebLogic Server 6.1 now transforms an XML document as expected (using a given XSL).
068656	Fixed a problem that was causing a <code>NullPointerException</code> in <code>weblogic.xml.jaxp.ChainingEntityResolver.pushEntityResolver</code> .
073311	Using the <code>weblogic.apache.xalan.xslt</code> to transform huge amount of data led to <code>javax.xml.transform.TransformerException: String index out of range</code> . This has been fixed.
075800	The built-in XML parser now correctly parses XML files that use external entity references to include external chunks of XML within the original XML file.

## Third-Party JAR Files in WebLogic Server 6.1 Service Pack 3

JAR File	Description	
J2EE, javax and subdirectories	JTA 1.0.1	
	JCA 1.0	
	JMX 1.0	
	EJB 2.0	
	Servlet 2.3	
	JSP 1.2	
	JMS 1.0.2	
	XML	org.apache.xalan 2.0.1
		org.apache.xerces 1.3.1
Xerces 1.2.0 - DOM and SAX parsers		
org.w3c.dom interfaces		
org.xml.sax interfaces		
LIBRARIES	antlr 2.7.0 - runtime classes of antlr compiler construction toolkit	
	bsh 1.2b3 - bean shell	
	certicom - SSLPlus/Java 3.1.12B (used by Node Manager)	
	netscape - Netscape Directory SDK 4.1 for Java	
	Oracle 8.1.6	
	Oracle 8.1.7	
	Apache Jakarta Ant - 1.3 Apache build system	
	Apache Jakarta Oro 2.0 - regular expression matching library	

# WebLogic Server 6.1 Service Pack 2 Solutions

The next sections describe problems that were resolved in WebLogic Server 6.1 Service Pack 2.

- EJB
- Examples
- JDBC
- jDriver
- JMS
- Miscellaneous
- Plug-Ins
- RMI over IIOP
- Security
- Servlets, JSP and Web Applications
- System Administration
- Web Services
- WebLogic Tuxedo Connector
- XML

## EJB

Change Request Number	Description
044294	Fixed a memory leak that could occur when creating multiple entity beans; the garbage collector was not reclaiming all of the used memory.
049590	The <code>ejbc</code> compiler now will produce IDL files in the correct order, putting primitive value types before non-primitives.
049709, 055644, 059540	<p>Previously, due to timing problems, exceptions/errors were thrown when EJBs were deployed in a cluster and then a re-deploy or un-deploy was attempted.</p> <p>It was possible that an undeploy JNDI message would arrive after local undeploy had taken place. This would cause unmarshaling errors, as the classloader for an EJB is discarded once local undeployment of an EJB is done. This unmarshal problem would result in <code>DistributedManagement</code> exceptions popping up in log files, and the EJB could never be un-deployed from a cluster.</p> <p>Similarly it was possible that a JNDI binding announcement would arrive before local deployment was successful, resulting in same unmarshaling/ de-serialization errors (as EJB classes required to unmarshal are not there yet).</p> <p>Fixes in 6.1 Service Pack 2 take care of both these problems and remove any timing dependencies in local deploy/undeploy and JNDI Announcements. Now you can successfully deploy/undeploy/redeploy EJBs in a cluster without any exceptions and problems due to race conditions.</p>
051454	In the case where an EJB has been idle longer than the value of <code>idle-timeout-secs</code> , it is removed from the cache and <code>ejbremove()</code> is called.
055113	The <code>examples\ejb\basic\statelessSession\build.xml</code> file had a typo where the name of the EJB jar file was misspelled. This has been fixed.
055163	Fixed a problem with stateful session beans that are configured with in-memory replication across a cluster. The bean was failing over when an <code>EJBException</code> was thrown, but not when the bean was removed.
055502	A message-driven bean now recovers messages after an application throws an exception.
055510, 056543	Fixed problems with container-managed persistence 1.1 EJBs not re-reading deployment descriptors after being redeployed.
057119, 054806	Fixed an out of memory problem that occurred when redeploying EJBs several times over; the <code>&lt;max-beans-in-free-pool&gt;</code> descriptor was being ignored and exceeded.

## 6 Resolved Problems

---

058210	The <code>ejbRemove()</code> method is now invoked at undeployment for both stateful session beans and stateless session beans with both home and remote interfaces.
058857	Fixed problem where entity beans weren't being recognized if the home didn't have a <code>create()</code> method.
059043	<code>ejbc</code> no longer complains when the <code>EJBLocalObject()</code> method has a throws exception clause that does not throw a <code>RemoteException</code> .
059046	Accessing a stateful session bean from a stateless session bean in a cluster no longer fails.
060018	Accessing a <code>ReadOnly</code> EJB with container managed relationships no longer causes a <code>LockTimeout</code> exception.
055842	Fixed a problem where a message-driven bean could not access a remote queue on another physical machine because it the client could not be authenticated.
058074	Storing a reference to an EJB local interface in an instance field in a stateful session bean no longer causes an assertion error during passivation.
053523 and 060032	Fixed a problem where the guess method <code>getURL()</code> in <code>weblogic.ejb20.internal.BaseEjbHome</code> was using a non-secure port when the protocol was <code>t3s</code> and <code>https</code> .
054335	Removing a <code>ReadWrite</code> entity bean now invalidates its corresponding <code>ReadOnly</code> entity. Previously, the <code>ReadOnly</code> entity would remain cached.
059674	Using a read-only bean-managed persistence entity bean no longer calls <code>ejbLoad()</code> twice when the bean has been loaded via a finder.
060142	Fixed a problem in the <code>getEJBHandle()</code> method such that failover is now working in a 2-node cluster.
060416	Fixed a problem where performing an EJB lookup in a tiered environment caused a <code>ClassCastException</code> to be thrown when the EJB Home object was cast to its specific type.
061883	Fixed a problem where reentrant calls to read-only beans failed because of a lock timeout.
062515	The <code>findByChar(char c)</code> method no longer fails for a container-managed persistence EJB home.

## Examples

Change Request Number	Description
055155	<p>A problem was fixed in the following sample file:  <code>ejb20\cascadeDelete\one2many\table.ddl</code></p> <p>The problem resulted in a “SQL not properly ended” error message (error number 933).</p>
055370	<p>Broken links are now fixed in the following examples under  <code>wlserver6.1\samples\examples\jsp</code>:</p> <p><code>SimpleSession.jsp</code>  <code>URLEncode.jsp</code></p> <p>For <code>SimpleSession</code> the link on Session Timeouts was missing.  For <code>URLEncode</code> the link Writing Web Application Deployment Descriptors was missing.</p>
056711	<p>Building the <code>wtc/simpFML32</code> example no longer generates a warning about container transactions.</p>
057237 and 056063	<p>The RMI-IIOP C++ examples have been fixed to work on UNIX platforms. They are located in the following subdirectories:</p> <p><code>/samples/examples/iiop/ejb/entity/tuxclient</code>  <code>/samples/examples/iiop/ejb/stateless/server/tux</code>  <code>/samples/examples/iiop/ejb/stateless/tuxclient</code> <code>/samples/examples/iiop/rmi/server/tux</code>  <code>/samples/examples/iiop/rmi/tuxclient</code></p>

## JDBC

Change Request Number	Description
055044	<p>Dynamically created connection pools are no longer treated as persistent within the <code>config.xml</code>.</p>

## 6 *Resolved Problems*

---

---

055435	The JDriver for Oracle now retrieves the correct number of 'null' values when it queries the database and the database has more than 500 rows.
056268	When trying to execute SQL queries with the language set to Russian, the use of hyphens in the timestamp no longer contain error causing hyphens.

---

## jDriver

---

<b>Change Request Number</b>	<b>Description</b>
046149	Fixed a problem where when WebLogic Server wrote a multibyte string to a CLOB or NCLOB column, it wrote an incorrect string length.

---

## JMS

---

<b>Change Request Number</b>	<b>Description</b>
055293	JMS default connection factories are now properly bound as a function of the <code>JMSDefaultConnectionFactoriesEnabled</code> property in the <code>config.xml</code> file. Previously, if no other JMS objects ( <code>JMSConnectionFactory</code> , <code>JMSServer</code> ) were defined on a given WebLogic Server, then the default connection factories were not bound into JNDI.
056320	Fixed problems in JMS with an error deserializing an object in the case of using a <code>Topic</code> to send a string as an <code>ObjectMessage</code> and then sending a <code>RemoteObject stub</code> as another <code>ObjectMessage</code> .
061783	Fixed a problem that occurred when trying to simultaneously deploy instances of different message driven beans that subscribe to the same JMS topic. Previously, one or both beans would stop working.

---

## Miscellaneous

---

<b>Change Request Number</b>	<b>Description</b>
036679	The Administration Console now functions correctly when resizing the Netscape browser frames on Solaris.

---

## 6 Resolved Problems

---

---

041873	Fixed problems with the <code>weblogic.policy</code> file when users started up the server with <code>-Djava.security.manager</code> . Even when the user had correct permissions, a <code>java.security.AccessControlException: access denied</code> exception was being generated.
041991	When attempting to get the <code>MBeanHome</code> from the <code>RemoteMBeanServer</code> , an application no longer receives "null".
043010	If WebLogic is running as a Windows service it will no longer shut down when you log out of the Windows session that the server is running in.
050437	When running under WebLogic Express, WebLogic no longer prints out: <code>&lt;Error&gt; &lt;JMS&gt; &lt;JMSServer "null", JMS license validation failed, com.bea.utils.misc.NoSuchProcessException: No such license: WebLogic Server 6.1, JMS.&gt;</code> . This message was misleading and harmless.
053732 and 054781	<p>Introduced a <code>-delay</code> option to the <code>beasvc.exe</code> utility which ensures that Managed Servers only start up provided that an Administration Server is already running. The new <code>-delay</code> option lets users delay the Managed Server's JVM thread execution for a specified number of milliseconds.</p> <p>Previously, a Managed Server failed to start up because it depended on a running Administration Server.</p> <p><b>Note:</b> For information about related changes in WebLogic Server 6.1 SP04, see "Change in Behavior of <code>-delay</code> Argument for <code>beasvc.exe</code>" on page 1-4.</p>
053990	The Node Manager will now load up servers that have whitespace in their names.
055007	WebLogic Server now implements a throttling mechanism to reject outstanding requests after a certain limit. Previously, the <code>acceptBacklog</code> value was not effective in limiting the number of new connections that could possibly overwhelm the server.
055442	Fixed a problem involving sending XML over HTTP. The problem was that WebLogic Server was erroneously calling the <code>getInputStream()</code> method.
056332	Fixed a problem causing an <code>IllegalStateException</code> when calling a RPC-style Webservice.
057091	Fixed a problem with applets in Internet Explorer, where 'didn't meet stated Content-Length' errors were being thrown because the content length check was being performed at the wrong time.
058832	Fixed a problem where <code>HttpSession</code> failover service failed and a session was lost if a client accessed the session just when a primary server was shutting down. This occurred only in a two-server cluster.

---

---

059119	<p>Fixed a problem where a Java client talking to a Weblogic cluster performed more slowly on a 3-server cluster than on a 2-server cluster. Moreover, performance would keep degrading with addition of more servers until the client ensured that there were at least as many socket reader threads on client as the number of servers in the cluster. This involved configuring Execute Thread count and Percent Socket readers client-side properties.</p> <p>This fix removes the dependency of such a configuration, as the number of servers can grow or shrink dynamically. Now, WebLogic Server creates new threads for socket reading on fly making sure there is one thread per outgoing socket from client. This means customers don't have to tweak those properties for the client for optimal performance when a WebLogic client talks to a WebLogic cluster.</p>
059219	<p>Fixed a problem encountered by wireless application where WebLogic was retrieving the port number from a location that could have incorrect information.</p>
060557	<p>The FileT3 attribute now creates a file with the correct path when the path is targeted to a managed server.</p>
061060	<p>The <code>getTimeout()</code> and <code>setTimeout()</code> methods are no longer missing from the class <code>URLConnection</code>.</p>
056911	<p>Previously, if an entry was added to the manifest Class-Path for a <code>.war</code> file, it would be loaded by the web applications's <code>ClassLoader</code>. With this fix, the application's <code>ClassLoader</code> now loads it. The net effect is that all web applications now share instances of classes loaded from the manifest Class-path, whereas before they all got their own copies.</p>
051220	<p>Fixed a problem where multi-byte system messages in WebLogic Server log or console were corrupted and, therefore, unreadable.</p>
054871	<p>The Administration Console now displays a JNDI tree correctly. Previously, deployments were appearing under the wrong context.</p>
057466	<p>Fixed a problem where executing the <code>weblogic.Admin PING</code> command to a server running on Solaris with Japanese locale (<code>LANG=ja_JP.PCK</code>) resulted in a <code>java.net.SocketException</code> displaying on the server.</p>

---

058786

The Netscape Enterprise Server Plug-In (NSAPI plug-in) enables requests to be proxied from Netscape Enterprise Server (NES, also called iPlanet) to WebLogic Server. The plug-in enhances an NES installation by allowing WebLogic Server to handle those requests that require the dynamic functionality of WebLogic Server.

Previously, reading the HTTP header in the NSAPI plug-in resulted in an unexpected EOF: “PROTOCOL\_ERROR [line *linenumber* of *filename*]: unexpected EOF reading HTTP headers at line *linenumber*.”

The underlying problem proved to be with parsing large cookies.

This problem is now fixed.

---

061460

You can now configure Weblogic 6.1 to restrict the number of client connections. There are two methods:

1. Edit the `config.xml` file to set the `weblogic.system.openSockCount` parameter, which limits the number of permitted connections:

```
<Server ListenPort="7001" Name="myserver"
NativeIOEnabled="true"
TransactionLogFilePrefix="config/mydomain/logs/"
MaxOpenSockCount=1000>
```

2. Use `weblogic.Admin` to set the socket count or write a JMX Java client to do so and utilize the API in `weblogic.management.configuration.ServerMBean`, as in this code fragment:

```
/**
 * Returns the maximum number of open sockets allowed
 * in server at a given point of time. When max threshold is reached,
 * server stops accepting any more new requests until no of sockets
 * drops below threshold.
 *
 * @default java.lang.Integer.MAX_VALUE
 * @legalMin 0
 * @legalMax java.lang.Integer.MAX_VALUE
 *
 * @configurable
 *
 * @oldprop weblogic.system.openSockCount
 */

int getMaxOpenSockCount();

/**
 * Sets the maximum number of open sockets allowed
 * in server at a given point of time. When max threshold is reached,
 * server stops accepting any more new requests until no of sockets
 * drops below threshold.
 *
 * @default java.lang.Integer.MAX_VALUE
 *
 * @legalMin 0
 * @legalMax java.lang.Integer.MAX_VALUE
 *
 * @configurable
 *
 * @oldprop weblogic.system.openSockCount
 *
 */

void setMaxOpenSockCount (int sockCount);
```

## 6 *Resolved Problems*

---

---

051882	If <code>web.xml</code> is missing from your web application and you attempt to edit or enter a deployment descriptor through the console, a blank screen no longer appears.
--------	--

---

---

## Plug-Ins

---

<b>Change Request Number</b>	<b>Description</b>
053641	Now the plug-ins correctly respond to the setting of the <code>KeepAliveEnabled</code> parameter. The <code>Connection: Keep-Alive</code> header was not being sent back to WebLogic when using HTTP 1.1.
055276	There are no longer problems when disabling caching with the ISAPI plug-in.
055750	Fixed problems using <code>MatchExpression</code> with Stronghold versions of Apache.
056628	Fixed a problem that occurred when setting the Apache HTTP Server Plug-In <code>PathTrim</code> parameter. Setting the <code>PathTrim</code> parameter to “/” no longer causes the loss of session information.
060064	Fixed a problem where installing the Apache HTTP Server Plug-In as a statically linked module on the Solaris platform generated fatal errors.

---

## RMI over IIOP

---

<b>Change Request Number</b>	<b>Description</b>
056233	<code>ejb11.security.HomeMethodPermissionTest.testHandleRemoveWithoutPermission()</code> no longer fails with an out of memory error.

---

## Security

---

<b>Change Request Number</b>	<b>Description</b>
060491	Fixed a problem where single sign-on (SSO) did not work when the initial resource in the web application was not a protected resource.

---

## 6 *Resolved Problems*

---

---

CR061313

Please review the security advisory information at

<http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-13.jsp>.

---

## Servlets, JSP and Web Applications

Change Request Number	Description
046879, 057371	A cgi executable now successfully serves an image to the browser (.gif, .jpg or .png format). Previously, attempting to do so resulted in an exception being thrown in the WebLogic Server log.
047698	The <code>IndexDirectoryEnabled</code> property now works correctly for managed servers: both directories and files will be listed. Previously, if <code>IndexDirectoryEnabled</code> was set to “true” for a particular web application and that web application was requested on a managed server, directories were not listed, only files. Also, if files were added or removed from the web application, the changes were not reflected when browser page was updated.
050402	Fixed a problem that occurred when redeploying a web application with a different name. Previously, when a web application was already deployed through the Administration Console and then mbeans were used to programmatically deploy the same .war application under a different name, WebLogic Server would go into an infinite loop.
053513	Fixed a problem where, when using <code>HttpClusterServlet</code> to proxy requests, a page sometimes displayed improperly in the browser.
053706	Fixed problems when <code>AuthFilter</code> is registered in a web application and <code>ServletAuthentication.weak</code> is called. Now, <code>doSuccessAuth</code> is correctly called in <code>AuthFilter</code> when this is the case.
053965	If characters between ASCII 128 and 156 are a part of a URL parameter which is then subject to a <code>response.sendRedirect</code> , the second servlet no longer receives those characters as ASCII 63 and the characters are sent to the second servlet correctly.
054206	<code>weblogic.servlet.internal.ServletRequestImpl</code> —the implementation class of <code>HttpServletRequest</code> —now properly handles “folded” HTTP header lines. Previously, code would be incorrectly parsed.
054211, 055987	You can now configure the lifetime of HTTP persistent connections, through the new client-side properties <code>http.keepAliveCache.lifeTime</code> and <code>http.keepAliveCache.proxyLifeTime</code> . Previously, the lifetime was hardcoded on the client side. This value overrode the server-side <code>keepAlive</code> value and led to scalability problems.

## 6 Resolved Problems

---

054852	<p>A web application with a configured error handler now redirects to this error handler when JSP1 forwards to JSP2 where JSP2 throws an exception.</p> <p>Previously, the server returned a blank page and showed a status code of 200. The server logged the exception, but the client did not see it.</p>
054898	<p>Fixed a problem with JSP pages using <code>Authfilter</code> and <code>doSuccessAuth</code>, where the first JSP page is in a series of JSP pages that are contained within different web applications.</p>
055333	<p>Fixed a JSP compilation problem when using a <code>contentType</code> tag in the middle of the JSP rather than in the beginning.</p>
055466	<p>Fixed a problem with servlet filters unable to call EJBs and causing <code>ClassCastException</code>s.</p>
055535	<p>Fixed a problem where, when using the JSP refresh tool, the file distribution servlet was copying files to the wrong location.</p>
055583	<p>When implementing <code>TryCatchFinally</code> in a JSP custom tag, the correct java code is now created upon compilation.</p>
055595	<p>WebLogic Server now resolves tag library URIs in JSPs.</p>
055636,56265	<p>When using a <code>getParameter()</code> method in a JSP, if the value entered is empty, null is no longer returned. Now an empty string is returned. This fixes certain problem with leaving out equal signs in a query string.</p>
056092	<p>The JSP compiler no longer complains erroneously of duplicate variable declarations.</p>
056189	<p>WebLogic Server now supports clients using “Transfer-Encoding: Chunk”, i.e., the client no longer needs to send the “Content-Length” Header if “Transfer-Encoding: Chunk” is used in the post request.</p>
056322	<p>Intermittent RMI errors are no longer thrown when two different servlets in separate web applications attempt to access the same EJB.</p>
056668	<p>When a parameter is empty, the <code>getParameter()</code> method now returns “empty” instead of “null”.</p>
056955	<p>Fixed a problem where the post data from a request was lost after a <code>forward()</code> (one servlet forwards the request with post data to another) when a web application had <code>&lt;charset-params&gt;</code> defined in its <code>weblogic.xml</code> or <code>&lt;context-param&gt;</code> defined in its <code>web.xml</code>, the post data from a request is lost after a <code>forward()</code>.</p>

057101	Fixed a problem where the <code>getParameter()</code> method was not returning the first of multiple values when including a JSP. Previously, the second value was being returned instead of the first.
057279	Fixed a discrepancy in behavior between exploded directory and packaged web applications. In an exploded directory scenario, classes were being loaded in the incorrect order with <code>WEB-INF/lib</code> classes being loaded before those in <code>WEB-INF/classes</code> .
057620	Special characters such as '+' and '&' are now decoded correctly when the codeset encoding is 'UTF8'.
057684, 057559	A <code>ClassCastException</code> no longer occurs when calling <code>PortableRemoteObject.narrow()</code> when an applet is a client to an EJB.
058125	The <code>client.jar</code> that is generated for web services is now compatible with JDK 1.2 java clients.
058187	Fixed a problem with <code>weblogic.deploy</code> where a JSP refresh was failing; a new directory added to an exploded directory application was not being propagated to the managed servers and hence refresh was failing.
058220	The servlet error <code>java.lang.NullPointerException</code> at <code>servlet.internal.ChunkOutput.clearBuffer</code> no longer occurs when trying to send a SOAP request.
058352	You can now access web applications with one-character names, such as "i.war." Previously, although such an application would deploy, access would fail with a 404 error.
058714	Fixed a problem where a <code>.gif</code> image was improperly displayed when streamed through a <code>cgi-bin</code> executable versus when the image was 'streamed' directly via the web container.
058738	The HTTP version is now maintained correctly in a response header when the request is sent via HTTP 1.0.
058931	Deleting a JSP now results in a 404 error instead of a 500 error.
059054	Fixed a problem where WebLogic Server failed to parse “,” separators in cookies, resulting in new sessions erroneously being created.
059180	Fixed a problem where WebLogic Server was overwriting an existing cookie created by another entity, rather than simply adding its own cookie.

## 6 Resolved Problems

---

---

059412	Fixed a servlet problem where if you called <code>session.removeAttribute()</code> twice, then called <code>session.setAttribute()</code> once, one <code>removeAttribute</code> remains rather than both removes being overridden.
060192	Fixed a problem where requests were not pinned to a primary server and sessions were lost when using URL rewriting to track sessions. (Cookies were disabled on the browser.)
060536	Fixed a problem where <code>jspc</code> was erroneously exiting with an exit code of <code>SUCCESSFUL</code> .
060595	Fixed a problem where WebLogic Server threw a <code>ClassCastException</code> with nested JSPs when using a filter that wrapped the <code>HttpServletRequest</code> .
060615	Fixed a problem where session ids were not consistent across web applications. Now, each web application sees a single, consistent session id for a given user's session.
060963	WebLogic Server no longer reuses a sessionid when the web application is not participating in single signon.
061277	(JSP 1.1) <code>javac</code> compile error messages are no longer corrupted on a Japanese platform.
061387	Fixed a problem where WebLogic Server created a new session every time the page was requested, when the <code>JSESSIONID</code> cookie was sandwiched between two user defined cookies.
061488	Fixed a problem where request parameters were parsed incorrectly when a "=" was used in the value.
066184	In 6.1 SP1, the default encoding used for servlets and JSPs was the JVM default. As of 6.1 SP2, the default encoding has been changed to <code>ISO8859_1</code> . This change is in accordance with the latest servlet specification. Servlet Specification 2.3 SRV 4.9 states that the default encoding should be <code>ISO8859_1</code> .

---

---

# System Administration

---

Change Request Number	Description
045780	<p>Multiple server definitions with the same name are now disallowed. If you have started a server and then attempt to start up the second server with the same name, the server startup aborts, with this error message:</p> <p>"Server: &lt;yourservername&gt; already exists, make sure that you do not have duplicate copies of this Server name in your XML config file"</p>
055190	<p>Fixed a problem where JSP refresh failed if the application was targeted to a cluster. Previously, you could not add new JSPs or change or delete existing JSPs from a running web application without having to undeploy and redeploy the application.</p>
056091	<p>A new procedure is available to restart an Administration Server on a different machine while Managed Servers are still running. For more information, see the section "<a href="http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#failover">Failover Considerations for the Administration Server</a>" at <a href="http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#failover">http://e-docs.bea.com/wls/docs61/adminguide/startstop.html#failover</a>.</p>
058946	<p>Fixed a problem where precompiled JSP's on a Managed Server would recompile even without if the pages hadn't been modified. As a result it took a long time to start up the Managed Server.</p> <p>With this fix, if classes are already compiled on an Administration Server, when the Managed Server starts up the classes are shipped and their timestamps are preserved, to the Managed Server will not try to recompile them.</p>
054624	<p><code>weblogic.deploy</code> now behaves as expected when its <code>-component</code> option is invoked.</p> <p>Previously, it did not behave as expected when using the <code>-component</code> option: the application was unintentionally deployed to a server that was not in the component list of servers.</p>
054573	<p>Fixed a problem where any configuration error involving a <code>JDBCConnectionPool</code>, <code>MultiPool</code>, or <code>(Tx)DataSource</code>, did not get reported in an error dialog in the console. Rather, it appeared to be correctly configured.</p>
058183	<p>Fixed a problem where WebLogic Server was deploying <code>.war</code> files before <code>.jar</code> files. At server startup, a web application would deploy before an EJB, resulting in a <code>NameNotFoundException</code>.</p>

---

## 6 Resolved Problems

---

---

059702	<p>Fixed a problem where web applications could not be deployed when network cards were disabled.</p> <p>Previously, WebLogic Server would try to validate with the DTD at <a href="http://www.bea.com/servers/wls610/dtd/weblogic-web-jar.dtd">http://www.bea.com/servers/wls610/dtd/weblogic-web-jar.dtd</a>. When it cannot reach <a href="http://www.bea.com">www.bea.com</a>, it failed. Now, when there is no network access, the server finds <code>weblogic-web-jar.dtd</code> in <code>weblogic.jar</code> and continues.</p>
060244	<p>Fixed a problem where, when using the Administration Console to target a cluster for just one application, individual servers were also pulled into the target list. With this fix, the cluster is no longer expanded to a list of servers.</p>
061642	<p>Fixed a problem with converting a 5.1 <code>weblogic.properties</code> file to a 6.1 <code>config.xml</code> file. The conversion utility was not converting the password for the <code>JDBCConnectionPool</code> correctly. As a result, attempts to start the server in the new domain failed.</p>
055067	<p>Fixed a problem that caused web applications that had dependencies on classes in <code>EJB.jar</code> files not to deploy when precompile was turned on.</p>
060411	<p>(EJB 1.1)</p> <p>Fixed a problem where a second call to a required EJB transaction method cause the transaction to roll back. The first call removed some entities and then created some. The second call removed the previously created entities and then created some new ones.</p>
042052	<p>Fixed a problem occurring in migration from one release to another: the <code>weblogic.properties</code> file was not converting completely, such that the <code>web.xml</code> file of the web application did not contain all of the properties.</p>
061300	<p>Fixed a problem where the WebLogic Server console was executing in the default queue (a lower priority queue) instead of “<code>__weblogic_admin_html_queue</code>” (a higher priority queue). This compromised the console’s ability to respond when the server was experiencing heavy load.</p>

---

## Web Services

Change Request Number	Description
055415	There is no longer an interoperability problem when using the SoapWare test specification and the UserLand client (reference implementation).
054749	It is no longer true that when using the <code>wsgen</code> Ant task to assemble a stateless session EJB into an rpc-style Web Service, <code>wsgen</code> fails with a parsing error if the package level of the EJB contains only one level, such as: <pre>package mytest;</pre>
055645	The <code>Remote2WSDL</code> class, used to assemble a WebLogic Web service manually, now correctly includes a <code>protocol</code> option for specifying a protocol other than <code>http</code> , such as <code>https</code> .
056332	Previously, if an exception occurred while WebLogic Server was processing a SOAP request to invoke a WebLogic Web service, the WebLogic Server executed the <code>ServletResponse.getWriter()</code> method twice. This in turn created another exception. This does not happen anymore.
056382	WebLogic Web services now support the <code>xsd:decimal</code> data type.
056631	WebLogic Web services now support the <code>xsd:hexBinary</code> data type.
056639	You no longer get an error when you use the WebLogic Web services client API to invoke an rpc-style WebLogic Web service whose stateless session EJB has security constraints defined for it in the <code>ejb-jar.xml</code> and <code>weblogic-ejb-jar.xml</code> files.
058230	The WebLogic Web services client API now correctly constructs a complex Java object on the client side after receiving the object inside a SOAP request. This complex object consists of an array of JavaBeans, one of whose attributes is itself an array of a different type of JavaBean.
058431	WebLogic Web services now correctly return nested array data type values.
059653	When using <code>Remote2Wsd1</code> to manually assemble Web services, you can now specify the <code>context</code> and <code>jndi_name</code> part of the URL equivalent to the <code>location</code> attribute of the <code>&lt;soap:address&gt;</code> element of the WSDL. This means that you can now successfully invoke a method on the Web service proxy obtained from the WSDL.

## WebLogic Tuxedo Connector

Change Request Number	Description
055889	A <code>TPReplyException</code> thrown with <code>TPESVCFail</code> from WebLogic Tuxedo Connector will now be received by the Tuxedo Client.
057150	<code>WTCValidateCF</code> utility no longer throws an error when validating a configuration file that has <code>FldTblClass</code> elements.

## XML

Change Request Number	Description
044211	The <code>xmlx.jar</code> file now correctly contains the <code>HTMLEntities.res</code> resource file.
046934	<p>The built-in SAX parser now correctly parses deployment descriptor files that are written using any character set.</p> <p><b>Warning:</b> Prior to this fix, WebLogic Server would have failed to detect encoding problems in your <code>.xml</code> files for English-only applications. Consequently, English-only applications that ran without fail in the past may now fail with this Service Pack 2 fix; WebLogic Server will report an encoding-not-supported error.</p> <p>If WebLogic Server reports this error, check the specified <code>.xml</code> file for the correct encoding name and syntax.</p>
055082	The <code>XSLTInfo.properties</code> file in the <code>weblogic.jar</code> archive is now correctly called <code>/weblogic/apache/xalan/res/XSLTInfo.properties</code> rather than the incorrect <code>/org/apache/xalan/res/XSLTInfo.properties</code> , causing transformations to work correctly and not throw a <code>SAXException</code> .
056839	The <code>weblogic.apache.xalan.serializer.Encodings</code> class now contains the correct mapping between MS932 and Shift_JIS.
062320	The WebLogic FastParser now parses very large strings (> 1MB) in seconds rather than minutes.

# **WebLogic Server 6.1 Service Pack 1 Solutions**

The next sections describe problems that were resolved in WebLogic Server 6.1 Service Pack 1.

- Deployment Descriptor Editor
- EJB
- Examples
- JDBC
- JMS
- JTA
- Miscellaneous
- Plug-ins
- RMI over IIOP
- Servlets and JSP
- System Administration
- Web Services
- XML

## Deployment Descriptor Editor

<b>Change Request Number</b>	<b>Description</b>
041479	Fixed a memory leak that was occurring during updates (redployment of) J2EE applications.
048757	Removed all yellow caution icons that were next to deployment descriptor editors on the console.
048804	The Container Transaction Trans Attribute no longer has the '(none)' option for the EJB deployment descriptor editor in the console.
049465	In the console EJB deployment descriptor editor, the display name for the Weblogic RDBMS Jar is derived from the filename where the descriptor was read.
051858	All RDBMS descriptors referenced in the <code>weblogic-<i>ejb-jar.xml</i></code> file are now available in the deployment descriptor editor.
051939	Fixed the <code>ejb-jar.xml</code> parsing code to accept <code>Auto-acknowledge</code> and <code>Dups-ok-acknowledge</code> as legal <code>acknowledge-mode</code> elements for Message-Driven Beans.
052613	The link to edit the deployment descriptor is now hidden during application configuration.
052665	If the <code>web.xml</code> file has a valid element called <code>&lt;run-as&gt;</code> , it is no longer lost when using the deployment descriptor editor.
052804	The <code>&lt;charset-params&gt;</code> entities and all child entities are now persisted to the <code>weblogic.xml</code> file by the deployment descriptor editor.
052963	Access to the deployment descriptor editor for <code>.war</code> , <code>.jar</code> , or <code>.rar</code> files no longer exists in the component table of the console.
053089	The deployment descriptor editor now indicates that the Primary Key that is used with the Automatic Key Generation feature must be of Java Type: <code>java.lang.Integer</code> .

---

053098	Added a legal values list to the MessageDrivenDestinationMBean's SubscriptionDurability attribute. Removed the SubscriptionDurability attribute from the MessageDrivenMBean and removed the SubscriptionDurability property from MessageDriven.jsp.
053282	Added the exclude-list tag to the EJB 2.0 deployment descriptor editor.
053375	The deployment descriptor editor is no longer blank after a modified bean is redeployed using the check box in the console.
054682	When persisting EJB deployment descriptor settings from the deployment descriptor editor, the contents of all description elements are now enclosed in CDATA sections to prevent possible parsing errors.

---

## EJB

<b>Change Request Number</b>	<b>Description</b>
031350	The EJB compiler now correctly validates that a primary key class for an entity bean has all public fields. Previously, the validation existed but was broken.
044294	Fixed a memory leak that was occurring when repeatedly creating entity beans.
047050	If a message-driven bean's <code>onMessage</code> throws an unchecked exception, rolling back the transaction is now called instead of <code>setRollbackOnly</code> .
048506	Fixed a memory leak where multiple beans were created and not removed.
048716	EJB compliance error messages now indicate which bean is faulty.
049180	<code>EntityPool</code> did not call <code>unsetEntityContext()</code> when asked to release a bean. This has been fixed.
049658	Fixed a problem with using arithmetic operators with input parameters in <code>ejb-ql</code> . Capability was added for these three symbols: <code>+ - /</code>
050134	The <code>weblogic.ejb</code> debug flag is now working. To compile with debugging, use <code>-g</code> instead of <code>-debug</code> . For example: <pre>java weblogic.ejb -g foo.jar bar.jar</pre>
050860	The <code>DDConverter</code> now generates <code>ejb-ql</code> with updated syntax for the <code>cmp20</code> bean.
051528	Added an optional <code>&lt;description&gt;</code> tag element in the <code>&lt;weblogic-query&gt;</code> tag in <code>weblogic-rdbms20-persistence-600.dtd</code> , so you can describe your query.
051653	Fixed a problem with SQL query generation in a many-to-one relationship where the primary key field is part of the foreign key.
051670	When an EJB that referred to another EJB's handle was passivated, a <code>ClassCastException</code> was thrown. This has been fixed.

051735	The EJB compiler was not recognizing the <code>-j</code> option when the compiler was specified as <code>sj</code> and hence was failing to execute the command. This has been fixed.
051803	The compliance checker now disallows <code>local</code> or <code>local home</code> to be passed as a parameter type or return type for the remote interface. See section 10.3.10.1 in EJB spec for more detail.
051807	The compliance checker no longer allows remote and local components to share the same CMR mapping.
051875	The XML generator no longer ignores values that are explicitly set. Even if the values are equal to their default values, they are generated.
051941	Fixed a problem with the EJB container that involved improper handling of a business method named <code>remove</code> .
052075	<code>SELECT DISTINCT</code> is now working when an <code>ejbSelect()</code> returns a collection of fields.
052502	Changes to EJB deployment descriptors are now picked up when EJBs are redeployed through the Deploy checkbox in the console.
053093	When the auto create tables feature is turned on, sequence tables are now generated.
053096	The <code>ejbSelect</code> method is now returning a set of local beans instead of returning a set of bean instances.
053101	Compliance checker no longer gives the wrong value for <code>&lt;acknowledge-mode&gt;</code> .
053287	The following entry in the <code>ejb-jar.xml</code> file no longer causes the entire file not to load: <pre>&lt;message-driven-destination&gt; &lt;destination-type&gt;javax.jms.Topic&lt;/destination-t ype&gt; &lt;subscription-durability&gt;Durable&lt;/subscription-d urability&gt; &lt;/message-driven-destination&gt;</pre>
053596	The Number of Active StatelessSessionBeans is no longer allowed to exceed the <code>max-beans-in-free-pool</code> setting.
053599	Improved the ability of the EJB <code>DDInit</code> tool to determine bean classes.

## 6 Resolved Problems

---

053643	If an EJB without a local interface calls <code>SessionContext.getEJBLocalObject()</code> , an <code>IllegalStateException</code> is now being thrown.
053646	When validating an <code>ejb-jar.xml</code> file, the compliance checker and XML validator now refer to “relationship-role-source” instead of “role-source” to comply with a change in the EJB specification.
053735	A null pointer exception is no longer thrown in <code>StatefulSessionManager</code> when invoking local method.
053768	Updated the deployer so that it does not look for the <code>weblogic.EJB20Enabler</code> class to enable EJB 2.0. Also, removed license error messages referencing a separate download because they are now irrelevant.
053801	A Compliance Checker error is no longer thrown when a message-driven bean's class does not contain an <code>ejbCreate()</code> method but its superclass does.
053827	The <code>DDConverter</code> utility can now be accessed using <code>java.weblogic.ejb20.utils.DDConverter</code> as documented.
053884	If an <code>ejbPostCreate</code> method was defined with a throws <code>CreateException</code> and if <code>SQLServer AutoKey Generation</code> was enabled, the generated code was incorrect. This has been fixed.
053905	When you use EJB2.0 BLOB/CLOB mapping on an UPDATE, if the UPDATED field was shorter than the original field, the newly UPDATED result no longer contains the left-over data from the original field.
053944	Fixed the exception handling code so that if a stateless session EJB's constructor was throwing a <code>RuntimeException</code> or <code>RuntimeError</code> , the caller's <code>java:comp/env JNDI</code> context was correct after accessing the EJB.
053970	Default role and relation names are now generated, making it possible to set up EJB 2.0 relationships without setting the <code>ejb-relation-name</code> element in the relation declaration of the <code>ejb-jar.xml</code> descriptor.
053986	The descriptor <code>mbean</code> is now consistent regarding XML case sensitivity.

---

054021	An assertion error is no longer thrown when a clusterable entity bean is called from a non-clusterable EJB.
054225	Fixed a problem with <code>ear</code> file deployment when it contains a circular reference in the <code>Class-Path</code> entry of the <code>Manifest.mf</code> file in jar files.
054449	The code generated for EJB 2.0 entity beans that use container-managed persistence was changed to work around a limitation in Oracle. A <code>cmp</code> -field that is mapped to an Oracle <code>LONGRAW</code> column can now have a length greater than 4096 bytes.

---

## Examples

Change Request Number	Description
045300	Added new JTA example that uses both Oracle's thin driver and JDriver. This example involves two database instances. For each database, the user must set up a single table and create a new transactional connection pool and a new transaction datasource. This example uses J2EE 1.3 features which use non-final API specifications.
048031	Clarified the instructions, syntax, and examples for specifying the cluster address and the listen port when invoking the example clients. Two files changed: <pre> ... \samples\examples\cluster\ejb\package.html ... \samples\examples\cluster\rmi\package.html </pre>
050429	Standardized table creation sql script to be vendor independent.
051476	A problem was fixed in <code>examples/ejb20/bands.demoPool</code> connection pool now points to <code>oraclePool</code> connection pool.
051491	Corrected the reference and example code in the <code>examples/jta/jmsjdbc/build.xml</code> file.
051714	A problem was fixed in <code>examples/ejb/sequence/mssqlserver/package-summary.html</code> . <code>demoPool</code> connection pool now points to an <code>mssqlserver</code> connection pool.
052595	The second compilation of the <code>i18n.logging.message</code> directory is no longer producing errors.
052689	Fixed a problem running more than one client using Petstore.
053008	Corrected information in the SSLClient example document. Now, there is no reference to a <code>production_apps</code> directory as the applications directory and, the <code>SnoopServlet</code> is located in the <code>/WLS_HOME/samples/examples/jsp</code> directory, not in the <code>/WLS_HOME/config/examples/jsp</code> directory.

---

053463	The logging internationalization examples no longer have broken links.
053466	<p>WL_HOME/samples/examples/i18n/logging/package-summary.html now contains a link to WL_HOME/samples/examples/i18n/logging/message/UsersMessagesServlet.html.</p> <p>The "Setting up your environment", "examplesweb application", and "Start the WebLogic Server with the examples configuration" links in WL_HOME\samples\examples\i18n\logging\message\UsersMessagesServlet.html now correctly go to WL_HOME\samples\examples\examples.html.</p>
053383	Added a set-up step that was missing from the JSP Form validator example.
053396	The wireless stockDemo example is now returning a live quote instead of an outdated quote.
053525	Starting the default server in mydomain in development mode was causing the Defaultweb application and certificate.war file in the applications directory to be deployed twice.
053852	Updated DOCTYPE to register latest version of DTDs.
053871	Modified build script to compile JSPs consistently across all platforms.
054963	The generated startscript for migrated domains is now consistent with other scripts.
056080	Fixed a problem that was resulting in the examples\i18n\logging\message build script failing.
056114	Added literal encoding to fix a problem with the message webservice sample.
056347	Petstore Shopping cart EJB no longer uses clustering.
056495	Removed the time example.

---

## JDBC

<b>Change Request Number</b>	<b>Description</b>
034041	Added the capability to rotate the JDBC log based on date. You can specify how often the log file should be rotated and when rotation should begin. The current <code>jdbc.log</code> file is has the most recent date timestamp at the end of the file name.
034046	It is no longer possible to create data sources based on multipools containing mixed XA and non-XA-aware connection pools because it could result in incorrect transactional behavior.
035380	It is now possible to dynamically create a JDBC DataSource using JMX APIs in addition to using the properties file.
042924	Creating a <code>TxDataSource</code> that points to a multipool now works. Previously, doing so resulted in a <code>ResourceException</code> .
045991	When you define a connection pool in the console by specifying a login and password as part of the URL instead of as properties, an exception is no longer thrown.
049963	Added a description to the javadoc comments for <code>JDBCConnectionLeakProfile</code> and <code>JDBCStatementProfile</code> interfaces that serve the connection pool runtime MBean.
051058	Changed checking of the "reserve" ACL to "admin" ACL in <code>weblogic.jdbc.common.internal.ConnectionPool.startup(Deployment MBean)</code> .
051165	Added reset methods to corresponding profile so that profile storage can be reset programatically to initial zero size.
051371	The <code>weblogic.Admin</code> JDBC usage message now includes connection pool commands.
052540	The size of the <code>PStatement</code> cache is now handled explicitly through the Administration Console and through the <code>JDBCConnectionPoolMBean</code> . Setting size of the <code>PreparedStatement</code> cache via database properties is deprecated.

---

052960	The getter and setter methods for <code>HighAvailability</code> and <code>LoadBalance</code> in <code>JDBCMultiPoolMBean.java</code> are now deprecated. They are replaced by the <code>AlgorithmType</code> getter and setter methods.
053085	Added synchronization to the <code>close()</code> and <code>isClosed()</code> methods in the JDBC pool driver's <code>Connection</code> class.
053721	The <code>getPoolList()</code> method still exists in <code>JDBCMultiPoolMBean</code> in 6.1 and is documented in the javadoc.
054187	Upon multiple invocations, the <code>PreparedStatement</code> query no longer returns an incomplete <code>ResultSet</code> .
055655	JDBC driver DLL for Oracle 8.1.6 is no longer broken.

---

## JMS

<b>Change Request Number</b>	<b>Description</b>
051938 & 053464	If a destination becomes unavailable, producers can now properly detect them as unavailable ( <code>peerGone</code> ) and will send and receive exceptions while the destination remains unavailable. Once the destination becomes available, producers will recover and resume working properly.
053635 & 054655	To get the proper transaction semantics, applications should always use an XA connection factory when inside an EJB. Furthermore, the default connection factory used by Message Driven Beans (MDBs) is now an XA connection factory. To configure user-defined connection factories used inside of an EJB as XA, use the Administration Console to select the XA Connection Factory check box ( <code>XAConnectionFactoryEnabled="true"</code> ) under the Configuration-Transactions tab under the Connection Factories node.
053649	WebLogic JMS was only acknowledging messages up to and including the message being acknowledged. It now acknowledges all messages (unless the Acknowledge Policy attribute is set to Previous).
054154	When a message is serialized outside of JMS (pass by value to an EJB or an RMI call), the destination is no longer dropped.
054365	The behavior of the pro-active assignment of messages performed by session pool connection consumers has been corrected. Regardless of the <code>messagesMaximum</code> setting on the connection factory, a connection consumer will consume at most the <code>connectionConsumer.messagesMaximum</code> setting to any given session.
055364	When attempting to set properties on a message that is received in read-only mode, users can now call the <code>clearProperties()</code> method to make the properties writable. Previously, if a received message had no properties to clear, <code>clearProperties()</code> would not make the properties writable.
056556	When multiple subscribers are created under the same session, asynchronous subscribers no longer hang after receiving the first message.

## JTA

Change Request Number	Description
024199	The JTA driver no longer requires read permission in the top level workspace.
048979	Fixed the JTA race condition that caused the following error if multiple clients used the same resources simultaneously: <code>Unable to create resource runtime mbean</code>
050789	The auto-update interval for picking up applications deployed in a domain appears under Domain options and no longer appears under JTA Subsystem.
050849	Improved JTA resource health monitoring. There is now a distinction between an unhealthy/blocked transaction and an unhealthy resource. Once a resource is marked dead, a general mechanism informs the resource provider so that it can refresh itself. Hard-coded constants, such as the two-minute interval (after which an unresponsive resource is declared unhealthy), are now configurable.
051631	Using prepared statements with <code>testConnectionOnRelease</code> results in an <code>SQLException</code> for the Weblogic OCI XA driver.
051673	Heuristic exceptions are no longer being reported inconsistently in single versus multiple server configurations.
053638	A work-around is provided for the Oracle XA driver limitation that results from accessing the same database instance from multiple servers.
054513	Fixed checked transaction behavior check.

## Miscellaneous

Change Request Number	Description
-----------------------	-------------

## 6 Resolved Problems

---

032447 & 054281	WebLogic Server no longer prints to the server log "Emergency," "Critical," and "Alert" messages that directly pertain to shutdown when the server is intentionally shut down by an Admin user.
047596	When the SHLIB environment variable is not defined, WebLogic Server no longer fails to find the performance pack when started.
048222	The console monitoring graphs are now working in Netscape 4.75.
048524	The <code>LoginContext</code> class always threw a <code>LoginException</code> with the message "Authentication Failed" regardless of the exception thrown by the underlying custom <code>LoginModule</code> implementation. The <code>LoginContext</code> class now propagates the exception thrown by the <code>LoginModule</code> implementation.
049668	The Service Pack installer can now delete empty directories.
050345	Now, there can be multiple versions of either the <code>il8n_user.properties</code> file or the <code>weblogic/msgcat/il8n.properties</code> file. This feature allows users to place their message information more flexibly and to update the system messages if necessary.
050385	Added the reset connection pool option to the console for Jolt connection pool. This option is also available in WebLogic Server 6.0 SP1 and SP2.
050489	It is now possible to pass an argument that contains a comment into the startup class without the parser splitting the value into two arguments.
050517	<code>OBJ_ADAPTER</code> is no longer thrown when a valuetype is passed to a CORBA server on Solaris.
050888	Fixed network classloading for classes stored in the server's classpath. Previously, a <code>ClassNotFoundException</code> was thrown in the client while the client unmarshalled objects of those classes.
051297	Improved the <code>ZipException</code> message when the application path is misconfigured.
052146	Fixed a problem with weight-based load balancing.

---

052349	Fixed a password-related problem that arose when migrating to WebLogic Server 6.1 from WebLogic Server 5.1 SP7 or later. Previously, if you had LDAP configured in <code>ldaprealm.properties</code> , after running the <code>weblogic.properties</code> file conversion, you had to add the password for the LDAP server to the <code>config.xml</code> file manually. This is no longer required.
052567	Added a detailed error message to be used if the performance pack is missing.
052666	Corrected the error message that relates to the <code>startNodeManager.cmd</code> script which is started from the <code>WL_HOME/config</code> directory. The error message now points to the correct directory.
053308	The Tuxedo server in the <code>simpappcns</code> example can now boot.
053320	Deleted these two examples: <code>time.ServerTimer</code> <code>time.ClientTimer</code> WebLogic Time is deprecated. Use <code>java.util.Timer</code> instead.
053422	Fixed a problem with the Deploy checkbox in the console. If you de-selected this checkbox, the application was not deployed, but the next time the server was booted, the application was deployed anyway.
053426	The conversion utility no longer creates and uses the <code>production_apps</code> directory.
053559	The <code>weblogic.ant.taskdefs.ejb[20].DDInit</code> tool is now better able to determine bean classes.
053584	Merged the Path field and the URI field used in the console when configuring a standalone component.
053617	Client-side rollbacks now release transaction data structures immediately.
053693	The <code>weblogic.net</code> API is now exposed in the <code>javadoc.packages</code> .
053725	The SNMP Agent screen in the console now has a field called "TargettedTrapDestinations" which is of type <code>SNMPTrapDestinationMBean</code> .

---

## 6 Resolved Problems

---

---

053952 and 054777	Fixed problems with starting Managed Servers from the command line. Also, the error log message did not indicate whether the error came before or from the current start-up instance. This has been fixed.
053955	Added <code>%JAVA_HOME\bin</code> to the <code>PATH</code> environment variable in the <code>startNodeManager.cmd</code> script and in the <code>startNodeManager.sh</code> script. Because of these additions, starting a Managed Server via the <code>NodeManager</code> no longer fails if the <code>NodeManager</code> has been started up without "java" in its <code>PATH</code> variable.
054131	In a cluster environment with a <code>ProxyServer</code> ( <code>HttpClusterServlet</code> ), when calls are made from a client through the <code>ProxyServer</code> , the nodes in the cluster now receive the true client details instead of the <code>ProxyServer</code> 's details.
054281	<code>ListenThread</code> no longer fails during <code>WebLogic Server</code> shutdown.
054517	Fixed an HTTP tunneling problem with making RMI calls through user-launched threads on the server.
054608	Added attributes to dialog and table pages of the console for the properties, <code>SNMPAttributeChange</code> and <code>SNMPLogFilter</code> . The attributes provide the option to enable a server for each property.
054673	The <code>snmpwalk</code> command, a command line utility, is now working correctly.
054734	Fixed problems that occurred with the connection pool after the resource adapter threw a <code>CONNECTION_ERROR_OCCURRED</code> exception.
054738	Added support to allow an <code>ear</code> to contain a <code>rar</code> but have the <code>rar</code> 's Java classes packaged in a <code>jar</code> that exists in the <code>ear</code> rather than in the <code>rar</code> itself. The <code>classpath</code> entry is now supported in the <code>rar</code> 's <code>Manifest.mf</code> file.

---

---

054771	<p>When the connection leak detector was in effect, <code>ManagedConnections</code> were automatically destroyed in the time specified by <code>connection-cleanup-frequency</code> and <code>connection-duration-time</code> elements in the <code>weblogic-ra.xml</code> deployment descriptor file. However, if the resource adapter itself recognized that the <code>ManagedConnection</code> was currently active, it had no way to abort the <code>ManagedConnection</code> destruction.</p> <p>The resource adapter can now abort the automatic destruction of a recognized active <code>ManagedConnection</code> by throwing <code>javax.resource.spi.IllegalStateException</code> from a <code>ManagedConnection.cleanup()</code> method call.</p>
054777 and 053952	<p>Fixed problems with starting Managed Servers with the command line. Also, the error log message did not indicate whether the error came before or from the current start-up instance. This has been fixed.</p>
054795	<p>The installer now only permits a single 30-day trial period for a given product or release, service packs included. A license to extend the initial period can be obtained from the BEA Web site.</p>
055090	<p>During various interactions with WebLogic Server, the resource adapter received <code>NoAccessRuntimeExceptions</code> exceptions, which prevented the resource adapter from performing various operations. These access problems have been fixed.</p>
055121	<p>The start script that is generated by the converter tool now quotes <code>%CLASSPATH%</code>.</p>
055681	<p>All <code>nav.jsp</code> files now send out the nav tree in UTF-8 encoding so that the JNDI tree on Solaris can be displayed properly.</p>
056481	<p>Fixed bean deployment path in the connector <code>config.xml</code> file. The <code>connector.ConnectorEJBTest.testOpenConnection</code> no longer fails.</p>

---

## Plug-ins

<b>Change Request Number</b>	<b>Description</b>
047784	The <code>iisforward.dll</code> file now supports multiple websites for the same IIS instance
050080	The <code>PathPrepend</code> and <code>PathTrim</code> parameters for the Apache plug-in can now work simultaneously when they use HTTPS over an SSL.
050253	The ISAPI plug-in now allows the content-length to be equal to 0 for POST and for PUT.
051912	The NSAPI plug-in ( <code>proxy35.dll</code> ) for Netscape Enterprise Server 3.5.1 on Windows NT was giving the error message, <code>ns-httd36.dll could not be found in the specified path</code> . Because there is no such DLL in NES Server 3.5.1, this error message has been deleted.
052444	When you use the <code>HttpClusterServlet</code> plug-in and the primary server shuts down, a more informative error message is now displayed.
052480	The ISAPI plug-in now flushes the buffer even when the request is not served completely.
053773	Added a plug-in option to specify to which IP address the plug-in should bind when it connects to WebLogic Server.
054140	Added a new parameter, <code>WLProxySSL</code> , for all of the plug-ins whose default value is OFF. This parameter indicates whether HTTPS is used by the users' browser on which the proxy is based. The proxy then determines to which URL to send users in the case of <code>response.sendRedirect</code> .
054526	Fixed problems that resulted from using the same socket for the keep-alive connections.
054998	Fixed a problem with downloading a file from WebLogic Server 6.1 through IIS Web Server 5.0 as proxy. The error saying that <code>inetinfo.exe</code> has crashed no longer occurs.
055093	Fixed a problem that resulted from using SSL with the IIS plug-in.

## RMI over IIOP

<b>Change Request Number</b>	<b>Description</b>
052086	Fixed a problem that arose from calling outbound using IIOP after re-deploying an EJB.
052300	Fixed a problem with ear classloaders that prevented call by reference from working.
054182	In the 6.1 release, there was an IIOP/SSL outage. This has been fixed. A convenience class for accessing the SSL socket factory in the client was added.
054376	Security credential propagation is now supported over IIOP.
055330	Corrected the IIOP JVMID discriminators that are generated for remote JVMs.
055394	Fixed the SVUID for <code>weblogic.iiop.IIOPRemoteRef</code> .
056505	Fixed a problem with encoding value types in Anys with the IBM JDK on AIX.

## Servlets and JSP

<b>Change Request Number</b>	<b>Description</b>
C043505	Please review the security advisory information at <a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA00-09.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA00-09.jsp</a> .
043659	Shortened the session ID for WAP in a cluster.

## 6 Resolved Problems

---

049139	<p>The following JSP no longer fails to compile.</p> <pre>&lt;%@ taglib uri="accents.tld" prefix="accents" %&gt; &lt;%@ page contentType="text/html; charset=ISO-8859-1" %&gt; &lt;accents:test&gt; fail &lt;/accents:test&gt;</pre>
049147	<p>When you start the default server, a warning message referencing the servlet 2.2 DTD no longer occurs.</p>
050343	<p><code>ResponseHeaders.writeHeaders()</code> generates an HTTP header for single byte strings only.</p>
050370	<p>Added the <code>PrefetchTag</code> class to the <code>weblogic-tags.jar</code> file.</p>
050398	<p>Added the <code>weblogicx.jsp.tags.PrefetchTag</code> class to the <code>weblogic-tags.jar</code>.</p>
051396	<p>Load-on-startup now produces a warning if an invalid value is entered.</p>
052109	<p>A <code>NullPointerException</code> is no longer thrown every time <code>ServletSessionRuntimeBeanImpl.getMainAttribute()</code> is called.</p>
052134	<p>Added support for two Web applications or two <code>ejb-jars</code> with the same name in different <code>ear</code> files. Previously, deploying two <code>ear</code> files that each contained a web application <code>foo.war</code> file caused an <code>InstanceAlreadyExists</code> error.</p>
052665	<p>Added the capability to configure a <code>&lt;run-as&gt;</code> element for a servlet from the web application deployment descriptor editor.</p>
052786	<p>Exception-type mapping implementation no longer looks at the exact match for the exception type. Multiple classes can now be registered in the hierarchy. Exception-type mapping picks up the closest match.</p>
053314	<p>Added performance optimizations for in-memory replication and memory session persistence.</p>
053333	<p>The JSP page is now used as an error page if an exception is thrown in the Web application.</p>
053344	<p>Fixed a problem caused by using multiple slashes in a URL.</p>

---

053377	<p>The exception <code>weblogic.servlet.internal.session.MemorySessionDataMissing no-arg constructor for class</code> is no longer thrown when:</p> <ul style="list-style-type: none"><li>• A servlet invoked an EJB method and passed it a reference to its <code>HTTPSession</code> object.</li><li>• The servlet and EJB had been deployed in the default Web application.</li><li>• Memory-based session persistence was used by the servlet.</li></ul>
053573	<p>When more than one call for <code>Forward()</code> of <code>RequestDispatcher</code> is made within a Servlet service method, the container now throws an exception instead of returning a blank response.</p>
053634	<p>The query parameters added in a forward or an include now take precedence over the existing ones, in accordance with the servlet specification.</p>
053722	<p>Introduced a switch in the <code>weblogic.xml</code> file that redirects the client using the absolute URL. For example:</p> <pre>&lt;container-descriptor&gt; &lt;redirect-with-absolute-url&gt;true&lt;/redirect-with-absolute-url&gt; &lt;/container-descriptor&gt;</pre> <p>If set to <code>true</code>, the absolute URL is used.</p>
053743	<p>Now, in accordance with the servlet spec, an <code>IllegalStateException</code> is thrown if <code>ServletA</code> tries to forward the same request to the JSP after having forwarded it to <code>ServletB</code>.</p>
053932	<p>The <code>HttpClusterServlet</code> no longer throws a <code>Null Pointer Exception</code> and <code>weblogic.utils.AssertionError</code> if <code>secureProxy</code> is on.</p>
053959	<p><code>BodyTags</code> can now return <code>EVAL_BODY_INCLUDE</code> from their <code>doStartTag</code> method.</p>
054016	<p>Fixed the response header format corruption problem when the <code>Content-Disposition</code> header is present.</p>

---

## 6 *Resolved Problems*

---

054194	The content in the response buffer is now flushed as soon as the amount specified by the <code>setContentLength</code> is fulfilled.
054427	Added a mapping for jnlp files to the <code>weblogic/servlet/internal/webapplicationServletContext.java</code> file.
054694	<code>RequestDispatcher</code> now invokes Servlet 2.3 filters.
055002	A file can now be downloaded, instead of only appearing in the browser, after setting up mime-type in the <code>web.xml</code> file.
055106	Parameters for the request are now returned after the <code>HttpServletRequest.getParameter (param)</code> method for a POST request is executed in the <code>doPost ()</code> method in a servlet.
055241	The <code>request.getAttribute</code> and <code>request.setAttribute</code> are no longer broken when <code>Content-Type=text/xml</code> and the method is POST.
055265	New parameters no longer appear lost when <code>getServletContext ().getRequestDispatcher</code> and <code>request.getParameter</code> are used in the arguments.
055591	A null pointer exception is no longer thrown in <code>request.getParameterMap ()</code> if the method is called with any of the <code>getParameter ()</code> methods.
055630 & 055962	Fixed a problem with internal <code>ServletResponse</code> implementation.
056188	Fixed a parse error. The JSP container was failing to parse when <code>pageEncoding</code> was omitted.
056277	There is no longer an <code>ArrayIndexOutOfBoundsException</code> thrown on POSTs when using the Netscape browser.
057022	The JSP compiler no longer generates a classpath that is too long for the Win32 command line when <code>Class-Path</code> is used.

---

## System Administration

<b>Change Request Number</b>	<b>Description</b>
045203	The message produced due to an incorrect password when using <code>weblogic.Admin</code> is now consistent regardless of the command used.
049177	Fixed an error related to Web application deployment in clusters.
049178	Fixed an error that occurred when deploying the default Web application.
049919	Redundant copies of <code>ServerConfigMBeans</code> are no longer created on the Administration Server and the Managed Server.
050374	Converting the <code>weblogic.properties</code> file no longer fails when <code>DeliveryMode=persistent</code> .
050853	Configuration and deployment can no longer take place with the wrong path and URI.
050874	A cloned server can now be started as a Managed Server.
051618	<code>weblogic.Admin</code> now fetches the information from <code>MBeanHome</code> of the server running at the address specified in the <code>-URL</code> option. If the <code>-URL</code> option is not specified, <code>weblogic.Admin</code> fetches information using the <code>AdminHome</code> of <code>Domain</code> .
052046	When a password is not specified using the <code>-password</code> option in the command line for <code>weblogic.Admin</code> , the user is prompted to enter the password.
052095	When launching the Admin startup script, the Administration Server was started but did not detect running Managed Servers. Now, when the Administration Server starts up, it always loads the root directory from the properties directory ( <code>-Dweblogic.RootDirectory</code> ) or uses the default directory (current directory).
052198	A cloned server no longer fails during start-up.
052261	Fixed a problem with using the <code>weblogic.deploy</code> utility to deploy multiple EJB targets.

## 6 Resolved Problems

---

052326	If an erroneous root directory is specified in the <code>config.xml</code> file, WebLogic Server gives a warning message and the configuration is persisted to the directory where the Server is booted.
052349	When the deployment descriptor converter is run to generate a <code>config.xml</code> file from the <code>weblogic.properties</code> file that has the property <code>weblogic.security.realmClass=weblogic.security.LDAPRealm</code> set, a Password attribute now gets created in addition to the ConfigurationData for the CustomRealm. Values for both ConfigurationData and Password are based on the information contained in the <code>ldaprealm.properties</code> file. When the <code>config.xml</code> gets saved, the Password gets encrypted.
052656	Applications in the auto-deploy directory are no longer deployed twice.
052753	Starting the Administration Server and then starting two Managed Servers simultaneously no longer causes a server failure.
052804	If the <code>&lt;charset-params&gt;</code> entry exists in the <code>weblogic.xml</code> , then the deployment descriptor editor displays it. Previously, it failed to display and gave a <code>ClassCastException</code> .
053053	Fixed a problem with redeploying a message-driven bean after undeployment.
053485	This is a known issue with the conversion utility for customers migrating from earlier releases of WebLogic Server as well as those users following the migration tutorial.  The JDK automatically specified by the conversion process under the <code>compileCommand</code> tag in the <code>weblogic.xml</code> is set to <code>JAVA_HOME\javac</code> . This setting is no longer missing the <code>bin</code> subdirectory, which is set to <code>JAVA_HOME\bin\javac</code> . A corrected version of the conversion utility has been posted at <a href="http://developer.bea.com/tools/techguides.jsp">http://developer.bea.com/tools/techguides.jsp</a> .
053926	When the Administration Console was used to convert <code>weblogic.properties</code> on <code>src510sp10_117sj</code> to the <code>config.xml</code> file on <code>src_ag_131sj</code> , the root directory was not converted. This has been fixed.
053980	Fixed a problem that involved deploying a Web application in a cluster that was using an MBean API.

---

053991	MBean name patterns specified by JMX can now query the MBean server and find MBeans.
054173	Fixed a problem with starting Managed Servers simultaneously in a clustered environment.
054485	Improved the accuracy of Managed Server startup URL error messages.
055150	Fixed serialVersionUID for outer and inner classes following <a href="#">Sun's recommendation</a> .
055945	A <code>ClassCastException</code> was caused by deleting MBean references. This has been fixed.

---

## Web Services

<b>Change Request Number</b>	<b>Description</b>
048800	Fixed a problem with array elements. Array elements are no longer in the SOAP-ENV namespace. The .NET array always ends up as a non-null array of zero elements.
51222	You can now use the WebLogic Web Service client API to invoke a Microsoft-hosted Web service, as long as you add the following lines to your Java client program: <pre>CodecFactory factory = CodecFactory.newInstance(); SoapEncodingCodec codec = new SoapEncodingCodec(); codec.writeQualifiedNames( false ); factory.register( codec );</pre>
053249, 053254	When a static Java client invokes a message-style Web service by sending or receiving an XML document, the client no longer receives an error due to the WSDL of the Web service not supporting the literal encoding style.
058431, 054687	Null exceptions are no longer thrown when you try to invoke a remote procedure call Web service from a static Java client by using an array on primitive integers.

## XML

<b>Change Request Number</b>	<b>Description</b>
046955	Loading an XML document with <code>xmlns:include</code> in it from a Web application was causing an SAX exception. This has been fixed. The path for the <code>xmlns:include</code> must be page-relative.
048165	It is now possible to circumvent caching, in general and on specific entities.
049063	After you create parser and entity XML registry entries in the Administration Console, the public ID, system ID, and root tag fields become read-only and you cannot modify their values. This is the correct behavior.
049066	Changes in the console are now propagated to the entity cache.

049082	Debugging is now supported through an attribute in the <code>ServerDebugMBean</code> .
049855	The <code>xsl:include</code> was leading to the wrong path for the include stylesheet when used in stylesheets that were loaded by JSPs that used the XSLT tag library. This has been fixed. The path for the <code>xsl:include</code> must be page-relative.
050469	The Administration Server now correctly converts <code>XMLRegistryEntryMBeans</code> to the new MBean formats, even if an XML registry is not assigned to the Administration Server.
050620	When an entity was resolved, the registry entry was called once to resolve and once to get the timeout interval for entering into the cache. These functions are now performed by a single call.
052258	There are now default JAXP factories for standalone java applications. If the system properties are not set on the command-line, the weblogic factories are instantiated instead of the Apache factories.
054332	Fixed a problem that was causing an error to occur when parsing a well-formed XML document with the default parser.
054670	The error message that the parser produced when parsing an XML file if the user specified an undeclared tag is now easier to understand.
055499	When the <code>cloneNode(true)</code> method of <code>weblogic.apache.xerces.dom.Node</code> is called on a root element that contains a namespace attribute (i.e. <code>xmlns="http://www.abc.com/xml";</code> ) an erroneous <code>DOMException</code> is no longer thrown.

## WebLogic Server 6.1 Release Solutions

The following sections describe problems that were resolved for WebLogic Server 6.1.

- Deployment Descriptor Editor
- EJB
- Examples

- Internationalization
- JCA
- JDBC
- JMS
- Miscellaneous
- Plug-ins
- Servlets and JSP
- Web Services
- XML

---

## Deployment Descriptor Editor

---

<b>Change Request Number</b>	<b>Description</b>
048382 & 049320	Fixed some problems with displaying the Connector Deployment Descriptor Editor in the Administration Console when clicking on the menu item “descriptor.”
048965	Fixed some problems with displaying the EJB, web application, and WebServices nodes in the Deployment Descriptor Editor of the Administration Console. An error message was displaying when these nodes were clicked.
048966	Fixed problems with displaying Security Role Assignments in the web application node of the Deployment Descriptor Editor. Clicking Security Role Assignment caused an <code>IllegalArgumentException</code> exception to display and an error message for the web application.
048971	Fixed some problems in the EJB 2.0 Deployment Descriptor Editor when clicking on existing Resource Reference instances.
048973	Fixed some problems in the EJB Deployment Descriptor with persisting EJB entities. Clicking on the Persistence tab sometimes caused an exception.
049458	Fixed problems cloning the following: WebLogic RDBMS bean, WebLogic RDBMS Jar, transaction isolation, session beans and entity beans.
049460	The EJB Deployment Descriptor Editor had problems creating description folders for new WebLogic EJBs. Also, it was not possible to configure a new entity descriptor, stateless session descriptor or stateful session descriptor for a newly configured/cloned WebLogic EJB. These problems have been fixed.
049413	Fixed problems with persistence in each of the Deployment Descriptor Editors.

---

### EJB

---

<b>Change Request Number</b>	<b>Description</b>
042794	Deploying Message-Driven Beans into a full cluster through the Console's "Clusters" panel was broken. This is now fixed.
046960	Fixed a known problem with two servers hanging when a distributed transaction involving message-driven beans (MDBs) spans both servers.

---

### Examples

---

<b>Change Request Number</b>	<b>Description</b>
041463	Using UNIX, the <code>config/petstore/StartPetStore.sh</code> script has been modified to use the Hotspot VM on HP. This modification is a workaround for a bug in the HP client VM.
047509	When running the RMI examples, it was necessary to reboot the server after deploying. Before rebooting, the server was throwing a <code>javax.naming.NameNotFoundException</code> . This has been fixed.
049011	Fixed some problems with the <code>build.xml</code> file in <code>WEBLOGIC_HOME/samples/examples/xml/sax</code> .
049264, 048678	The PetStore demo now logs you out correctly when you do "sign out".

---

### Internationalization

---

<b>Change Request Number</b>	<b>Description</b>
035036	When WebLogic Server boots and prompts for the boot account's password, the prompt used is now internationalized.

---

---

041021	Fixed a problem failing to generate properties files for some message catalogs after executing l10ngen.
--------	---

---

041078	Fixed some duplicated keys in the <code>english.xml</code> file.
--------	--

---

041264	Fixed some internationalization problems with certain tags not being in <code>english.xml</code> .
--------	--

---

041266	The time stamp is now internationalized.
--------	--

---

### JCA

---

<b>Change Request Number</b>	<b>Description</b>
049346	WebLogic Server now supports "javax.resource.spi.security.GenericCredential" credential-interface or the "Kerby5" authentication-mechanism-type. Previously, specifying either value for the <authentication-mechanism> in the ra.xml file for the resource adapter being deployed resulted in a failed deployment.

---

### JDBC

---

<b>Change Request Number</b>	<b>Description</b>
033423	There were dbKona problems with QueryDataSet and the OCI805_8 driver. This has been fixed on Windows NT, but is still a problem on HP-UX and Solaris.
033920	There was a problem in JDBC when running tests which access the connection pool. This has been fixed.
035530	There were problems enabling OS authentication and using Oracle 8.1.6/oci8 with the jDriver. This has been fixed on NT and HP-UX, but still is an issue on Solaris. We have added an error message that reports this problem: OS level authentication is not currently supported due to a defect in OCI 8 libraries.
037589	Added Oracle thin driver support and fixed some outdated command line help.
037591	Added support for ORACLE_THIN database type. Removed obsolete database types and command line options.
037596	Fixed PreparedStatement.setBlob and PreparedStatement.setClob.

---

037693	There are no longer problems with the Oracle driver and ROWID selects when running <code>examples.dbkona.rowid</code> .
037893	Fixed <code>getBytes()</code> to return null instead of a zero-length byte array with Oracle LONG RAW columns.
038143	JDBCConnectionPoolRuntimeMBean's were not working. Added a RuntimeMBean implementation to <code>ConnectionPools</code> .
038229	The <code>jdbcDriver</code> for Oracle 8.1.6 no longer causes Managed Servers to run out of memory.
038275	We no longer refer to the <code>weblogic.classpath</code> in JDBC error messages. The <code>weblogic.classpath</code> does not exist in WebLogic 6.1.
040379	<p>The following error occurred when the <code>server</code> and <code>serverName</code> properties in a JDBC connection pool were specified with the same value:</p> <pre>&lt;Cannot startup connection pool "jtaXAPool" server and serverName properties must have the same value&gt;</pre> <p>This problem occurred when the XA driver required the <code>server</code> or <code>serverName</code> property to be set. It was occurring with the WebLogic <code>jdbcDriver</code> for Oracle, but not the Oracle thin driver, or the Cloudscape driver. It has been fixed.</p>
040509	Fixed a small problem with Solaris 2.6 and the <code>jdbcDriver</code> .
041286	Better error message for misconfigured connection pools.
041302	The MS SQLServer driver has been updated.
041514	Added support for JDBC and connection pool monitoring in the Console.
048764	Killing connection pools through the console is fixed.

## JMS

<b>Change Request Number</b>	<b>Description</b>
039726	If a pre-6.0 WebLogic Server was shut down when it had active database operations, after upgrading to 6.1 some durable consumers may have received messages that were sent to the topic after the durable subscriptions were created. There was a race condition between the scavenger (which updates the database periodically) and Ctrl+c killing the pre-6.0 server. This has been fixed.
039809	The JMS messages types, TextMessage, ObjectMessage, MapMessage and XMLMessage now implement in read-only mode so that the receiver cannot modify the message before calling clearBody().
041031	The JMS flow control no longer stalls asynchronous messages from a consumer.
041165	If a user receives a JMS ObjectMessage and calls the toString method before calling getObject, the message is now automatically deserialized.
041821	The JMS connection is now cleaned up properly when the client terminates without closing.
042096	Calling a setRollbackOnly in an MDB no longer yields a verbose log message.
042458	JMS can now handle sending messages that are received from a foreign provider.
042792	JMS can now process multiple messages concurrently with the JMSServerSessionPool.
042997	Server session pools no longer deadlock on multi-CPU boxes.
043077	The message listener no longer fails to auto acknowledge messages in server session pools.
043155	JMS topic messages now expire since they are checked at the time the message is given to the consumer.
043223	JMS can now send a TextMessage with more than 32K characters.

043225	When performing a setObject, the JMS ObjectMessage is longer unnecessarily unserialized.
043445	Sending a WebLogic Server JMS message using Fiorano 5.0 no longer generates an exception.
043447	When a JMS server session pool is shut down, its server sessions are now cleaned up and closed.
043519	A session rollback and recover no longer prevents asynchronous consumers from receiving messages.
043752	Recovering expired messages no longer causes a NullPointerException.
043884	<p>JMS provides better performance by not synchronizing writes to the file stores. Synchronous writes are configurable via the command line using properties and are enabled by default.</p> <pre>Dweblogic.JMSFileStore.SynchronousWritesEnabled=(false true)</pre> <pre>Dweblogic.JMSFileStore.&lt;store name&gt;.SynchronousWritesEnabled=(false true)</pre> <p>The first property applies to all JMS file stores running on the server. The second more specific property takes precedence over the first. The order where the properties are specified makes no difference.</p>
043973	TopicSession.unsubscribe() is now rejected if a message has been received as part of a transaction or has not been acknowledged in the session.
044200	The JMSConnection Mbean destroy() function now destroys the Connection when run on the server side. After destroying the Connection, the consumer is able to retrieve messages and the producer is able to send messages.
044285	JMS persistence mechanism no longer locks up when there are not enough server threads.
044450	The JDBC Store code no longer prepares "COMMIT WORK" unnecessarily.
044605	The serialization of an unsent JMS message no longer generates a NullPointerException.

## 6 Resolved Problems

---

---

044891	On a JMS transacted session, <code>Session.close()</code> now rolls back the in-progress transaction instead of committing it.
044961	The <code>getBooleanProperty()</code> correctly returns false if a value for name is not found. All of the other <code>getProperty</code> methods return <code>NumberFormatException</code> if a value is not found.
044976	<p>The returned values and exceptions from <code>getXXX()</code> in <code>MapMessage</code> now return the correct values, as follows:</p> <p>If the value for the field does not exist, then <code>getBytes()</code> should return <code>NumberFormatException</code>.</p> <p>If the value for field can not convert, then <code>getBytes()</code> should return <code>MessageFormatException</code>.</p> <p>If the value for the field does not exist, then <code>getBoolean()</code> should return false.</p> <p>If the value for the field can not convert, then <code>getBoolean()</code> should return <code>MessageFormatException</code>.</p>
045531	A <code>NullPointerException</code> no longer occurs when a JMS durable subscription is configured but a store is not configured.
045956	The connection consumer now behaves properly when creating a server session pool that is using default factories.
045963	The connection consumer no longer ignores the <code>maxMessages</code> field that caused it to swallow all the selected messages.
045980	A detailed log message was added for failures of the first SQL query, which describes the likely problem and provides a work-around. The work around ensures that the connection pool being used has permission to access the given tables. It also fully qualifies the table names by configuring a table name prefix in the JMS JDBC store that includes the schema and catalog names: ( <code>[schema.[catalog.]]prefix</code> ).
046705	When setting the message listener on a subscriber, JMS now remembers the context classloader at the time of setting so that when it calls <code>onMessage</code> on that message listener it can set the context classloader appropriately. This allows EJB to send messages to a web application subscriber that contains classes only present in the EJB.

---

---

046726	<p>There is a new configuration option for “acknowledge” to specify the new or old behavior. This has been added because JavaSoft has proposed changing the meaning of “acknowledge” based on the description in the JMS 1.0.2 specification as opposed to the one specified in the Javadoc, as follows:</p> <p>“Section 4.4.13: Acknowledging a consumed message automatically acknowledges the receipt of all messages that have been delivered by its session.”</p>
047424	<p>JMS no longer throws a NullPointerException when the message size exceeds the quota size.</p>
048181	<p>When creating a temporary topic, and then closing the connection, the following exception is no longer thrown when looking up the topic name: <code>weblogic.jms.common.JMSEException: Error unregistering RuntimeMBean.</code></p>
049518	<p>Fixed a problem migrating text messages generated using WebLogic Server 6.0 to WebLogic Server 6.1. The messages could not be received and the following message was thrown:</p> <p>The corresponding file block will be zeroed out on disk, and this corruption will no longer be reported in the future.</p> <p><code>java.io.StreamCorruptedException: Unknown object stream version.</code></p>

---

## Miscellaneous

<b>Change Request Number</b>	<b>Description</b>
035039	When WebLogic Server needs to prompt for the boot password, it now logs (in <code>security.xml</code> ) that it is prompting. This allows an administrator on a remote machine to know why the machine is apparently "hung".
036141	A Managed Server did not start if <code>guest</code> had been disabled (in order to do a JNDI lookup in an ACL). This has been fixed.
036604	The method <code>ClusterMBean.setDefaultLoadAlgorithm()</code> no longer accepts illegal values and now throws the proper <code>IllegalArgumentException</code> .
036986	Using Weblogic Server's XA driver with CMP no longer fails at deployment time.
038883	Specifying the root directory when starting the server was not working when the root directory was not under a <code>/config</code> directory. This has been fixed.
039066	If you entered an incorrect password when prompted at system startup, the server tried to start anyway. The startup eventually fails. Now you are re-prompted for the password when the password is wrong.
039182	When binding two clusterable objects implementing the same interface to the same JNDI name on a single server, an exception was not being thrown and the first clusterable object was being bound. This has been fixed.
039249	A Managed Server had a memory leak when it monitored JMSServer using the Administration Console. This has been fixed.
039361	If you create a group in the RDBMS realm, you can now delete the group.
039891	The server now sets the context attribute correctly during servlet initialization.
039960	Log Viewer is now working in non-U.S. locales.

039995	A server startup error no longer occurs after a WLEC Connection pool was created using the Administration Console.
040168	The logging example for internationalization failed when a dot (.) occurred in the path. This has been fixed.
040177	Additional user information was required when converting <code>weblogic.properties</code> files for use with WebLogic Server 6.0. This has now been added to the product documentation.
040178	When you were using the conversion utility, if a <code>weblogic.properties</code> file included the property <code>weblogic.httpd.register.Name=weblogic.servlet.FileServlet</code> and the servlet did not have initial arguments associated with it (it did not have the property <code>weblogic.httpd.initArgs.file=arg1=value</code> defined in the same <code>weblogic.properties</code> file), then the conversion utility was throwing a Null pointer exception. This has been fixed.
040213	The listenport of a Managed Server cannot be the same as the Administration Server listenport. You cannot, for example, use localhost or 127.0.0.1, with the 7001 host number on both the Managed and Administrative Servers. A more descriptive and informative error message has been added.
040297	Fixed a bug that prevented attributes set in <code>application.xml</code> from propagating from Administration Servers to Managed Servers.
040405	The <code>config.xml</code> file is now serialized using the Apache XML serialize classes. This results in the correct XML, including escapes for built-in entities and an XML header with encoding specified. The encoding used is the platform default. This also fixes problems with the “&” character.
040659	When specifying <code>getParameter()</code> encoding as iso-8859-1, the POST method is now working properly.
040688	When a mail session was targeted to a server that is part of a cluster, the following exception was being thrown because the mailsession was not serializable: <code>java.io.NotSerializableException: javax.mail.Session.</code> This has been fixed.

## 6 *Resolved Problems*

---

040698	The Application Manager no longer throws an unnecessary exception if the poller tries to deploy an application that is still in the process of being copied.
040899	Fixed a problem with a start-up class that occurred when the converter utility converted startup <code>weblogic.properties</code> into <code>.xml</code> .
040950	Fixed a problem in JTA. For XA connections, the Connection Pool test connection logic was attempting to start a distributed transaction before executing the testing SQL. This had the potential to create a <code>javax.transaction.NotSupported</code> exception.
041157	The dialog tabs in the Console were not working correctly when the Mbean name contained Kanji characters.
041290	If the username property is set from the <code>weblogic.Admin</code> or <code>weblogic.Server</code> command lines, the username must be the system password. Note however, that the username syntax is retained to prevent breaking a user's scripts. The <code>weblogic.management.password</code> property value must be the system password and the username may not be modified in the <code>config.xml</code> file.
041662	To improve JTA, we now advertise resources on JNDI. Resources were dynamically discovered as transactions go along, but now it is also possible to find out which resources are currently being supported at which servers.
041757	Fixed a problem with JTA. Resource advertisement (JNDIAdvertiser) was incorrectly done under the caller's identity. We now use the system identity while doing this operation.
041845	Updated the NTRealm to improve general usability and functionality.
041854	To help support Java WebStart with WebLogic Server, we now provide a default web application with the <code>web.xml</code> configured for this mime type. We have also provided other mime type mappings.
041995	Fixed an Unexpected Error being received in the console when resuming an active server. Also deleted some extraneous informational messages in the console when adding users.
042183	When the <code>config.dtd</code> was being generated in the console it was appearing with some errors. This has been fixed.

---

042362	According to the JTA spec documentation of <code>setTransactionTimeout()</code> : "seconds: The value of the timeout in seconds. If the value is zero, the transaction service restores the default value." If the value was zero, WebLogic Server was not resetting the default value. This has been fixed.
042794	Deploying message-driven beans into a full cluster through the Console's Clusters panel was broken. It is necessary to specify each server individually as a target through the Servers panel.
043369	Applications deployed across a domain now keep their original packaging, meaning that if an application was exploded on the Administration Server, it will stay exploded on the Managed Server.
043388	Clusters of WebLogic Servers no longer require the use of a shared network drive. Both the WebLogic Server installations and your applications can now reside on local file systems.
049417	The Password Converter Tool shipped with the Beta Release has been fixed and no longer returns a SAX Parser error for all files.

---

## Plug-ins

<b>Change Request Number</b>	<b>Description</b>
038831	The Apache plug-in no longer fails under heavy load.
040817	When using IIS to browse and call a regular JSP page that calls a Java Bean and getting a ResultSet reference, there were problems when submitting through the FORM POST method. The HTML stream was getting truncated when it came back to IIS--in other words, the page was only partially painted. This has been fixed.
041332	Using plug-ins, <code>MatchExpression</code> can now contain multiple, comma-separated expressions.
041534	Using the <code>mod_wl.so</code> , Apache was not proxying without <code>Debug</code> and <code>DebugConfigInfo</code> in the <code>httpd.conf</code> . This has been fixed.
041580	The Apache plug-in <code>MatchExpression</code> now works if an encoded URL is used.
041745	Added a log message when the trusted CA file isn't specified and SSL is used with the Isapi plug-in.
041754	Fixed an infinite loop when ISAPI was configured for SSL but using a non-secure port.
041939	The Apache plug-in wasn't filtering out the header <code>"transfer-encoding"="chunked"</code> . So the browser was failing to handle these responses, causing Netscape to hang and IE to display an error page. This has been fixed.
042482	The NSAPI and Apache plug-ins no longer have a READ timeout on HPUX-11.
044988	Fixed a problem with the NSAPI plug-ins where the socket was not properly closed by the plug-in side. The un-closed socket on the webserver side will be left in <code>CLOSE_WAIT</code> state.
046018	There are new binaries for Apache 2.0.
046392	It is now possible to have different cookie names set for different <code>&lt;Location&gt;</code> blocks when using the Apache plug-in.

046423	Weblogic Server now supports static libraries for Apache.
046424	<code>KeepAliveEnabled</code> and <code>KeepAliveSecs</code> parameters have been added for the Apache 2.0 plug-in.
046546	Cookie name is now configurable for persistent sessions. There is a new parameter, <code>PersistentStoreCookieName</code> (default= <code>WLCOOKIE</code> ), in <code>weblogic.xml</code> .
046946	<p>Fixed a problem with large deployments of WebLogic Server, wherein for a period of time a cluster node behaves as if it is the only one member in the cluster when it's not. Now, using <code>ListenDelaySecs</code>, you can delay the http request handling until it finds the other members. For example:</p> <pre>&lt;Server&gt; ListenDelaySecs="30" &lt;/Server&gt;</pre> <p>Delay the http listen thread for a configurable amount of time so that the new node has time to acknowledge the existence of other cluster members.</p>
046956	Resolved some problems caused by <code>DefaultFileName</code> and <code>VirtualHost</code> with Apache Stronghold/Raven proxies.
047987	Resolved a problem that occurred with NSAPI when streaming a large number of bytes (500,000 bytes) using SSL from the browser to the proxy and from the proxy to the server.
047636	WebLogic Server now supports DSO as well as statically linked modules for Apache 1.3.X.

## RMI-IIOP

## Servlets and JSP

Change Request Number	Description
026488	Fixed a problem with servlets when the parameter for method <code>println(String)</code> was null.
029989	The wrong error message is no longer sent when an <code>UnavailableException</code> was raised within a servlet.
030880	Fixed problems compiling JSPs when debugging was turned on. Now you can specify compiler flags in <code>weblogic.xml</code> . For example: <pre>&lt;!-- weblogic.xml: weblogic-specific web app descriptor --&gt; &lt;weblogic-web-app&gt; &lt;session-descriptor&gt; &lt;/session-descriptor&gt; &lt;jsp-descriptor&gt; &lt;jsp-param&gt; &lt;param-name&gt;compileCommand&lt;/param-name&gt; &lt;param-value&gt;jikes&lt;/param-value&gt; &lt;/jsp-param&gt; &lt;jsp-param&gt; &lt;param-name&gt;compileFlags&lt;/param-name&gt; &lt;param-value&gt;+E -nowarn&lt;/param-value&gt; &lt;/jsp-param&gt;</pre>
037771	Optimized the checking of JSPs for reloading.
038739	Improved support for JSP debugging.
039083	The <code>RequestDispatcher</code> and forwarding a JSP page now works correctly.
039193	When users forgot to make their servlet class public, a vague and unhelpful error was being thrown. This no longer happens.
039530	Fixed internal servlet tags to reset their state on release. If you used the same tag more than once on a page the additional tags kept the stale data from the previous tag.
040171	You can now set the attribute <code>SingleThreadedServletPoolSize</code> through the Administration Console.

---

040338	All functioning servlets are now displayed in the mydomain>web applicationlications>Defaultweb application_myserver>ActiveServlets window of the console.
040662	When JSP files had a page tag, the JSP compiler was not generating a class, or java file. This happened only with command line <code>jspc</code> . This has been fixed.
040766	Fixed a problem in servlets with using <code>../</code> for a relative URL when doing a <code>response.sendRedirect</code> .
040814	Added more informative error messages for instances when servlet user errors cause exceptions.
040843	Fixed problems with deploying web applicationComponents using virtual hosts.
040981	Fixed problems associated with WAP and session cookies using the pipe characters.
041234	Fixed a problem in JSP/Servlets using POST and HTTPS.
041478	A change to session cookie format fixed some problems with URL rewriting.
041605	JSPs compiler error pages no longer display a corrupt error message from <code>javac</code> .
041684	Fixed a problem in servlets. The XSLT tag threw a <code>SAXParseException</code> while processing XML content.
041729	The <code>weblogic.servlet.jsp.Precompiler</code> now honors <code>weblogic.xml</code> parameters, like <code>workingdir</code> , <code>packagePrefix</code> , etc., for the JSPs that it precompiles.
041945	When the <code>web.xml</code> had the setting <code>inputCharset./A*=Shift_JIS</code> <code>inputCharset./B*=EUC-JP</code> , and two clients simultaneously accessed the JSP/Servlet, WebLogic Server failed to handle these string correctly. It was possible to get the correct result from one client, but not from the other. This has been fixed.
042031	In the web application data table, when you click the monitoring icon for any web application and then click on the monitor link in the Servlets column, the correct heading is now displayed.

---

## 6 *Resolved Problems*

---

---

042293	Servlet URL rewriting was working fine with IE 5.0 but wasn't working with Netscape 4.7. This has been fixed.
042509	Fixed problems with forwarding from one web application to another web application and using authentication in both. Session data is no longer sharing inappropriately across the applications.
044873	When you compile a JSP manually with <code>weblogic.jspc</code> , the <code>-version</code> argument is now working.
044926	WebLogic Server 6.1 now correctly generates Java code from JSPs that include 2Bytes code.
049760	The <code>weblogic-vtags.jar</code> file containing the JSP tag library for Form Validation Tags is now in the <code>ext</code> directory as stated in the documentation.

---

## Web Services

Change Request Number	Description
046032, 046808	The WSDL of a WebLogic Web Service is now generated by a JSP, which by default dynamically sets the host and port of the WebLogic Server instance that hosts the service. You can specify that the host and port be hard-coded in the WSDL by explicitly specifying their values when you assemble the Web service using the <code>wsgen</code> Ant task.
046215	When you assemble a WebLogic Web Service using the <code>wsgen</code> Java Ant task, you can now specify that you want to use HTTPS (rather than the default HTTP) when invoking the Web service.
047794	When you use the <code>wsgen</code> Java Ant task to assemble a WebLogic Web service, the generated Enterprise Application archive no longer contains both a web application directory and a web application <code>*.war</code> file; it now contains just the <code>*.war</code> file.
048797	WebLogic Web Services now interoperate correctly with Array types in the Microsoft SOAP Toolkit.

## XML

Change Request Number	Description
034785	WebLogic Server now supports namespaces in the XML registry.
034785	Provided methods to retrieve namespace information about the document, such as <code>localRoot</code> , <code>qualifiedRoot</code> , and <code>namespaceURI</code> . If the XML document has namespace defined, then the lookup in the registry will be based on the qualified name.
036428	Previous releases of the XML subsystem included an error in Apache's implementation of the <code>javax.xml.*.parse(File)</code> method, included in the <code>xmlx.jar</code> file. Because WebLogic Server no longer includes this JAR file in the product and WebLogic Server's own implementation of the method has always been correct, this problem does not apply to the current release.

## 6 Resolved Problems

---

036642	<p>Generated custom parsers no longer convert newline characters that are embedded in attributes and tag bodies into a single space character.</p> <p><b>Note:</b> The ability to generate custom parsers is deprecated in this release of WebLogic Server. Instead, use WebLogic Server's high-performance parser.</p>
039096	<p>You no longer have to restart WebLogic Server to make it aware of a newly added entry to the XML registry.</p>
039247	<p>Custom parsers are now correctly invoked when an XML document that has been configured to be parsed by the custom parser is parsed using the <code>setAttribute()</code> method of the Servlet API.</p>
040479	<p>Fixed a bug in <code>JMSHelper</code> that was causing incorrect XML to be written to the <code>config.xml</code> file.</p>
041540	<p>The Xerces parser serializer bundled with the WebLogic server now supports multi-byte encodings, such as <code>Shift_JIS</code> or <code>EUC-JP</code>.</p>
041541	<p>The Xerces parser method <code>apache.xml.serialize.BaseMarkupSerializer.printEscaped()</code> bundled with the WebLogic server now supports multi-byte encodings such as <code>Shift_JIS</code> or <code>EUC-JP</code>.</p>
041734	<p>WebLogic Server's JAXP implementation no longer requires the input stream to be resettable when it contains an XML document that is about to be delegated to a particular parser for parsing.</p>
041778	<p>A <code>NullPointerException</code> error is no longer thrown when an XML document that contains an external entity that has been correctly registered in the XML registry is parsed.</p>
041797	<p>After you create a new server, the server no longer fails to reboot and the <code>config.xml</code> file no longer has some incorrectly parsed headers.</p>
46951	<p>The method <code>weblogic.xml.sax.XMLInputSource.getNamespaceURI()</code>, part of WebLogic Server's extension to the <code>org.xml.sax.InputSource</code> class, is implemented in the current release.</p>
48809	<p>The <code>SaxParserFactory</code> is now being selected correctly when parsing a document from a SAX Version 2 Servlet attribute.</p>

---

48810

You can now configure a parser for a particular document type by mapping the class name of the parser to a Public ID. NOTE: Specifying the parser class name when configuring a parser for a particular document type is deprecated in this release. Instead specify the Factory name.

---

