bea®

BEAWebLogic Server™ and WebLogic Express™

Release Notes

Contents

About This Document

1. WebLogic Server 7.0 Features and Changes

2. Known Issues

3. Resolved Problems for Service Pack 7

4. Resolved Problems for Service Packs 1 - 6

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 7.0 Features and Changes," is an overview of important information about WebLogic Server 7.0 and accompanying service packs.

- Chapter 2, "Known Issues" is a list of known problems in WebLogic Server 7.0 and its service packs.

- Chapter 3, "Resolved Problems for Service Pack 7" is a list of the software and documentation issues resolved in WebLogic Server 7.0 Service Pack 6.

- Chapter 4, "Resolved Problems for Service Packs 1 - 6," is a list of the software and documentation issues resolved in WebLogic Server 7.0 Service Packs 1 through 5.

Audience

This document is written for all users of WebLogic Server 7.0.

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 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.

Convention	Usage
Ctrl+Tab	Keys you press simultaneously.
italics	Emphasis and book titles.

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.
	
	Examples:
	`import java.util.Enumeration;`
	`chmod u+w *`
	`config/examples/applications`
	`.java`
	`config.xml`
	`float`

monospace italic text	Variables in code. *Example*: `String CustomerName;`
UPPERCASE TEXT	Device names, environment variables, and logical operators. *Examples*: LPT1 BEA_HOME OR
{ }	A set of choices in a syntax line.

[ ]	Optional items in a syntax line. *Example*:
	
	`java utils.MulticastTest -n name -a address`
	`      [-p portnumber] [-t timeout] [-s send]`

|	Separates mutually exclusive choices in a syntax line. *Example*:
	
	`java weblogic.deploy [list|deploy|undeploy|update]`
	`      password {application} {source}`

Convention	Usage
. . .	Indicates one of the following in a command line: • An argument can be repeated several times in the command line. • The statement omits additional optional arguments. • You can enter additional parameters, values, or other information
. . .	Indicates the omission of items from a code example or from a syntax line.

WebLogic Server 7.0 Features and Changes

Welcome to BEA WebLogic Server 7.0 Service Pack 7! As the leading Web application server, WebLogic Server™ implements J2EE 1.3 technologies, Web services, and other leading Internet standards to provide a reliable framework for highly available, scalable, and secure applications. WebLogic Server's seamless integration of disparate, heterogeneous platforms and applications enables your network to leverage existing software investments and share the enterprise-class services and data that are crucial to building mission-critical E-Business applications.

WebLogic Server documentation, including these release notes, is updated frequently at http://edocs.bea.com.

For deprecated features and APIs, see Deprecated APIs and Features in the *WebLogic Server 7.0 Upgrade Guide*.

The following sections describe the new features and major improvements made in the WebLogic Server 7.0 general release and its associated service packs.

What's New in WebLogic Server 7.0 SP7

WebLogic Server 7.0 SP7 offers the following important enhancements.

Administration Console

When WebLogic Server was executed as a Windows Service, applications located in remote mapped drives could not be deployed from the Console.

The set new Location option has been added to the *Locate Application* or *Component to Configure* page to enable you to deploy applications from a mapped network drive on Windows, using the UNC path.

Plug-Ins

- Support for handling chunked requests in NSAPI plugin has been added.

  Sun One web server will not de-chunk the incoming request only if the ChunkedRequestBufferSize property is set to 0 in the magnus.conf file.

- Using the IIS plugin and downloading large static files hosted on WebLogic Server was very slow. For files larger than 20 MB, several TCP packets were being sent to the client leading to performance degradation.

  The response buffer size has been increased to reduce the number of partially filled TCP packets.

What's New in WebLogic Server 7.0 SP6

BEA continues to support WebLogic Server 7.0 customers by providing WebLogic Server 7.0 service packs that resolve problems and introduce new features in WebLogic Server.

Administration Console

The Servlet Extension Case Sensitive attribute has been added to server and cluster configurations. In addition, the Web App Files Case Insensitive attribute has been added to security domain configuration.

These attributes specify whether file lookups for Java Server Pages (JSPs) are case sensitive on all platforms except win32; file lookups from standard win32 file systems are always case-insensitive. On case-insensitive file systems other than win32 (such as NT Samba mounts from UNIX or Mac OS that have been installed in case-insensitive mode), specify case insensitive lookups by setting these attributes to false to prevent the JSP from returning its source code. For example, if a JSP is being served from a Samba mount and you have specified case-insensitive lookups, WebLogic Server converts all file name extensions to lower case before looking up the JSP.

For more information, please review the security advisory at
http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp.

What's New in WebLogic Server 7.0 SP5

WebLogic Server 7.0 SP5 offers important enhancements.

Oracle 10g JDBC Thin Driver

In WebLogic Server 7.0SP5, the Oracle 10g (10.1.0.2.0) version of the Oracle Thin driver was added to the release and is now the *default version* of the Oracle Thin driver. See "Using Third-Party Drivers with WebLogic Server" in *Programming WebLogic JDBC*.

Note: The following are known issues with the Oracle 10g Thin driver:

- When using the multibyte character set with Japanese characters, a call to the `setString()` method fails. (Oracle TAR 3584729.994, Oracle Bug3437365). The version of the Oracle 10g Thin driver bundled with WebLogic Server 7.0 SP5 includes a patch for this issue. (WebLogic Server 7.0 SP5 is certified with the Oracle10g driver that is bundled with the kit and includes this patch.) If you are using the GA version of the Oracle 10g Thin driver, BEA strongly encourages you to obtain the related patch from Oracle.

- The Oracle 10g Thin driver has a batch size limit of 1682 operations. In previous versions of the Oracle Thin driver, there was no apparent size limit. If your application runs very large batch operations, you may see batch failures. To work around this issue, limit your batch size to 1682 or use the 9.2.0 version of the Oracle Thin driver. Oracle recommends batch sizes of between 5 and 30.

JDBC MultiPool Failover Enhancements

In WebLogic Server 7.0SP5, the following enhancements were made to JDBC MultiPools:

- Connection request routing enhancements to avoid requesting a connection from a disabled connection pool within a MultiPool.

- Automatic failback on recovery of a failed connection pool within a MultiPool.

- Failover for busy connection pools within MultiPools with the High Availability algorithm.

- Failover callbacks for MultiPools with the High Availability algorithm.

- Failback callbacks for MultiPools with either algorithm.

See "MultiPool Failover Enhancements" in *Programming WebLogic JDBC* for more details.

JDBC Connection Pool Testing Enhancements

In WebLogic Server 7.0SP5, the following features were added to JDBC connection pools to improve the functionality of database connection testing for pooled connections and to minimize delays in connection request handling:

- `CountOfTestFailuresTillFlush`—Closes all connections in the connection pool after the number of test failures that you specify, to minimize the delay caused by further database testing.

- `CountOfRefreshFailuresTillDisable`—Disables the connection pool after the number of test failures that you specify, to minimize the delay in handling connection requests after a database failure.

To enable these features, you add the attributes to the JDBCConnectionPool object in the config.xml file. Both attributes also require that the `TestConnectionsOnReserve` is set to `true` and a value is provided for `TestTableName`. See "JDBC Connection Pool Testing Enhancements" in *Programming WebLogic JDBC* for more information.

EJB Cache Size Trimming

You can now configure automatic trimming of idle read-write entity beans from your EJB cache. Automatic trimming prevents significant increases in the baseline memory footprint.

You configure cache size trimming by specifying the `idle-timeout-seconds` element with a value greater than zero. If no specification is made or the specified value is 0, automatic cache size trimming is not active and idle beans will not be periodically removed from the cache. For individual entity bean caches specified via the `entity-cache` tag in the `weblogic-ejb-jar.xml` deployment descriptor, the existing `idle-timeout-seconds` element will be enabled for those entity beans whose `concurrency-strategy` is set to Database, ReadOnly or Optimistic.

In addition to being applicable for individual entity caches via the `entity-cache` tag, this feature is also available for application caches where the `idle-timeout-seconds` is set via the `entity-cache-ref` tag.

Ant Tasks for Domain Configuration

In WebLogic Server 7.0SP5, the following Ant tasks were added to provide additional administration functionality for developers:

- `wlserver`—for creating new single-server domains and starting Administration Servers

- `wlconfig`—for configuring WebLogic Server domains

- `wldeploy`—for deploying applications to WebLogic Server domains

See Using Ant Tasks to Configure a WebLogic Server Domain in the *Administration Guide* for information about `wlserver` and `wlconfig`. See wldeploy Ant Task in *Developing WebLogic Server Applications* for information about `wldeploy`.

Configurable Log Rotation Criteria

Starting in Service Pack 5, if you have WebLogic Server installed as a Windows service, you can change the default time interval or use other criteria to archive the standard out and standard error logs. See Log File Rotation in the *Administration Guide*.

Enhanced Server Logs

You can configure the Administration Server to emit log messages when a user changes the configuration or invokes management operations on any resource within a domain. For example, if a user disables SSL on a Managed Server in a domain, the Administration Server emits log messages.

These messages provide an audit trail of changes within a domain's configuration (configuration auditing).

Client Tracking

Beginning with Service Pack 5, it is possible to monitor the IP addresses of durable subscribers. This information is now available on the `JMSConnectionRuntimeMBean`.

Custom Auditing Provider

Beginning with Service Pack 5, the default security realm for WebLogic Server includes a WebLogic Auditing provider. The WebLogic Auditing provider records information for any changes to the WebLogic Server domain configuration. This includes attribute values that have been changed, edited, or removed, as well as operations that have been invoked. See Do You Need to Develop a Custom Auditing Provider in *Developing Security Providers for WebLogic Server*.

New Security API

Previously, although you could access the `PrincipalAuthenticator.authenticate()` method by calling `weblogic.security.services.Authenticate.login()`, there was no way to access the `PrincipalAuthenticator.assertIdentity()` method, which would be useful in situations where the user has received the token other than in an HTTP or HTTPS header (for example, in an argument to a fat client). A new API, `weblogic.security.services.Authentication.assertIdentity()`, enables you to access `PrincipalAuthenticator.assertIdentity()`.

User Configuration and User Key Files for weblogic.Admin and Ant Tasks

For any `weblogic.Admin` command that connects to a WebLogic Server instance, you must provide user credentials. You can now use the new `STOREUSERCONFIG` command to encrypt the user credentials instead of passing credentials directly on the command line or storing unencrypted credentials in scripts.

Pausing and Resuming Message-Driven Beans

It is now possible to pause and resume MDBs.

WebDAV Support

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:

DELETE

GET

HEAD

OPTIONS

POST

PUT

*COPY

LOCK

MKCOL

MOVE

PROPFIND

PROPPATCH

SEARCH

UNLOCK

What's New in WebLogic Server 7.0 SP4

WebLogic Server 7.0 Service Pack 4 (SP4) coincides with WebLogic Platform 7.0 SP4, which optimizes Service Pack content across all WebLogic Platform components. Customers using WebLogic Integration 7.0, WebLogic Portal 7.0, or the full Platform 7.0 and requiring a service pack update beyond SP2 should use Platform 7.0 SP4.

What's New in WebLogic Server 7.0 SP3

WebLogic Server 7.0 SP3 offers important enhancements.

WebLogic Server 7.0 Service Pack 3 and WebLogic Workshop

WebLogic Server 7.0 Service Pack 3 (SP3) includes service pack updates for Weblogic Server and WebLogic Workshop. A WebLogic JRockit SP3 is also available and can be used in conjunction with WebLogic Server 7.0 SP3.

WebLogic Server 7.0 SP3 is intended for WebLogic Server and WebLogic Workshop support customers. WebLogic Server 7.0 SP3 is available through upgrade installers from the BEA support site (`http://support.bea.com`). The WebLogic Server 7.0 SP3 upgrade installers will not allow SP3 to be applied to an existing WebLogic Integration 7.0, WebLogic Portal 7.0 or full WebLogic Platform 7.0 installation.

BEA continues to support Platform 7.0 customers by providing Platform 7.0 service packs that incorporate service pack updates for all Platform 7.0 components (WebLogic Server, WebLogic Workshop, WebLogic Integration, WebLogic Portal, and Weblogic JRockit).

JDBC Vendor Connection Access from Pooled Connection

With Service Pack 3, you now have access to the underlying JDBC vendor connection from a database connection in a WebLogic connection pool. Some vendor-specific JDBC extensions require access to the vendor connection. There is a performance cost when using a vendor connection because once the vendor connection is exposed, the pooled connection is never returned to the connection pool for reuse. BEA recommends that you do NOT use a vendor connection unless absolutely necessary.

See "Getting a Physical Connection from a Connection Pool" in *Programming WebLogic JDBC*.

XA JDBC Statement Caching Enhancements

The prepared statement cache feature was changed so that the cache behaves differently for connection pools that use an XA (transaction aware) JDBC driver to create database connections instead of a non-XA JDBC driver. For the XA prepared statement cache, WebLogic Server uses a least recently used (LRU) algorithm to determine which statements to store in the cache for each connection in the connection pool.

See "Increasing Performance with the Prepared Statement Cache" in the *Administration Guide*.

JDBC Connection Leak Detection Enhancements

Enhancements were made to the JDBC connection leak detection feature so that when an application does not properly close a database connection from a JDBC connection pool (this is known as leaking a connection), Weblogic Server automatically creates a file named *server_name*slcn.tsf in the domain directory. The file lists the JDBC connection pool name from which the connection leaked and the stack trace where the connection was not closed. For any subsequent connection leaks, WebLogic Server appends the information to the file.

Support for Oracle Virtual Private Databases

WebLogic Server now provides support for Oracle Virtual Private Databases (VPDs). A VPD is an aggregation of server-enforced, application-defined fine-grained access control, combined with a secure application context in the Oracle 9i database server.

For more information, see "Programming with Oracle Virtual Private Databases" in *Programming WebLogic JDBC*.

JTA Transaction Log Write Policy

In WebLogic Server Service Pack 3, transaction log file write options were added to enable you to select the way entries are written to the transaction log. The option you select can affect server performance during transaction processing. Transaction log file write policy options are:

- Cache-Flush—(the default) Flushes operating system and on-disk caches after each entry to the transaction log. Transactions cannot commit until the commit record is written to stable storage.

- Direct-Write—Forces the operating system to write transaction log entries directly to disk with each write.

See "Setting the Transaction Log File Write Policy" in the *Administration Guide*.

Support for XAResource Transaction Timeout

The WebLogic Server Transaction Manager now supports setting a transaction branch timeout value on a participating XA resource if the resource manager supports the `javax.transaction.xa.XAResource.setTransactionTimeout()` method. You may want to set a transaction branch timeout if you have long-running transactions that exceed the default timeout value on the XA resource.

For the WebLogic Server Transaction Manager to set the transaction timeout on a JDBC XA resource, specify a value for the following properties in the JDBC connection pool tag in the `config.xml` file:

- `XASetTransactionTimeout`—A boolean property. When set to true, the WebLogic Server Transaction Manager calls `XAResource.setTransactionTimeout()` before calling `XAResource.start`, and passes either the `XATransactionTimeout` or the global transaction timeout in seconds. When set to false, the Transaction Manager does not call `setTransactionTimeout()`. The default value is false.

- `XATransactionTimeout`—The number of seconds to pass as the transaction timeout value in the `XAResource.setTransactionTimeout()` method. When this property is set to `0`, the

WebLogic Server Transaction Manager passes the global WebLogic Server transaction timeout in seconds in the method (see "JTA" in the *Administration Console Online Help*). The default value for this parameter is 0. If set, this value should be greater than or equal to the global Weblogic Server transaction timeout.

These properties apply to connection pools that use an XA JDBC driver to create database connections only. They are ignored if a non-XA JDBC driver is used.

When these values are set, the WebLogic Server Transaction Manager calls XAResource.setTransactionTimeout() as described above. The implementation of the method in the XA resource manager (for example, an XA JDBC driver) or the XA resource determines how the value is used. For example, for Oracle, the setTransactionTimeout() method sets the Session Timeout (SesTm), which acts as a maximum idle time for a transaction. The behavior may be different for other XA Resources.

The XASetTransactionTimeout and XATransactionTimeout properties are not available in the Administration Console. You must add them to the config.xml file while the domain is not active. For example:

```
<JDBCConnectionPool
  DriverName="oracle.jdbc.xa.client.OracleXADataSource"
  Name="oraclePool"
  Password="{3DES}8YdvP4FQW3k="
  Properties="user=SCOTT"
  URL="jdbc:oracle:thin:@server:port:sid"
  XASetTransactionTimeout="true"


XATransactionTimeout="120"/>
```

Removing Passivated EJBs from Disk

A new weblogic-ejb-jar.xml deployment descriptor element— session-timeout-seconds—specifies how long the EJB container waits before removing an idle stateful session EJB from disk. See session-timeout-seconds in Programming WebLogic Enterprise JavaBeans at http://e-docs.bea.com/wls/docs70/ejb/DDreference-ejb-jar.html#session_timeout_seconds.

context-root Naming for Web Applications

When you deploy a Web application as part of an Enterprise Application (.EAR file or directory), the Web application component is named after the context-root value specified in the

`application.xml` deployment descriptor. This value takes precedence over any `context-root` value specified in `weblogic.xml`, or the Web application's URI.

Controlling Access to the WSDL and Home Page of a Web Service

You can now control access to the WSDL and Home Page of a WebLogic Web Service by specifying the `exposeWSDL` and `exposeHomepage` attributes of the `<web-service>` element of the `web-services.xml` deployment descriptor file.

See Securing the WSDL and Home Page of a Web Service at http://e-docs.bea.com/wls/docs70/webserv/security.html#secure_wsdl.

Changes to Security Realm Options

Changes were made to the security realm options in this release of WebLogic Server. The Ignore security data in deployment descriptors option was replaced by the Check Roles and Policies and Deployment Descriptor Security Behavior options. The Administration Console online help does not describe these new options. Refer to *Creating a New Security Realm* at http://e-docs.bea.com/wls/docs70/secmanage/realm.html#new_security_realm in *Managing WebLogic Security* and *Techniques for Securing URL and EJB Resources* at http://e-docs.bea.com/wls/docs70/secwlres/types.html#techniques in *Securing WebLogic Resources* for a description of these options.

What's New in WebLogic Server 7.0 SP2

The following sections contain important notes for BEA WebLogic Server 7.0 SP2.

- "Extra EJB Compiler Options at the Server Level" on page 1-15

- "New Features for Installing WebLogic Server as a Windows Service" on page 1-16

Upgrade Installer Caveats

Note the following about the WebLogic Server 7.0SP2 upgrade installer:

- Remove from the CLASSPATH any references to 7.0SP1 patch files, as 7.0SP1 patch files are not compatible with 7.0SP2.

- The upgrade installer will not update any user-created files, such as a patch directory or user-created domains.

Migrating Domains Created Using the Configuration Wizard

The Configuration Wizard is a WebLogic Platform tool that allows you to create new domains quickly and easily. If you have domains that you created with the Configuration Wizard in a WebLogic Server 7.0 or 7.0 SP1 environment, and you want to continue running them in a WebLogic Server 7.0 SP2 environment, you must migrate them to SP2 before you can use them.

If you are migrating domains from WebLogic Server 7.0 to WebLogic Server 7.0 SP2, follow the instructions for migrating 7.0 domains to 7.0 SP1 domains in "Migrating Domains Created Using the Configuration Wizard" on page 1-20. If you have WebLogic Workshop domains, you must also follow the procedure in "Migrating a WebLogic Workshop Domain from SP1 to SP2" on page 1-12.

If you are migrating domains from WebLogic Server 7.0 SP1 to SP2, you must run the migration scripts as described in "Step 1: Upgrade Product JAR Files" on page 1-20. If you have WebLogic Workshop domains, you must also follow the procedure in "Migrating a WebLogic Workshop Domain from SP1 to SP2" on page 1-12.

Note: In addition to migrating domains to SP2, the migration scripts also upgrade the Sun SDK 1.3.1_03 to Sun SDK 1.3.1_06. If the Sun SDK is not used in your environment, modify your scripts appropriately.

Migrating a WebLogic Workshop Domain from SP1 to SP2

If you have a domain that you created with WebLogic Workshop 7.0 or 7.0 SP1, then you must revise the startWeblogic script for your Workshop domain as follows:

1. Add:

   ```
   call%WL_HOME%\common\bin\commEnv.cmd
   ```

You can add this line anywhere before the `set JAVA_DEBUG` statement. For new domains created with the Configuration Wizard using the WebLogic Workshop template, the `startWeblogic` script includes the call immediately following the lines that set the `WL_HOME`, `BEAHOME`, and `JAVA_HOME` variables. For example:

```
set WL_HOME=c:\bea\weblogic700
set BEAHOME=c:\bea
set JAVA_HOME=c:\bea\jdk131_06

CALL%WL_HOME%\common\bin\commEnv.cmd
```

2. Replace:

```
set JAVA_DEBUG=-hotspot
```

with the following line:

```
JAVA_DEBUG=%COMM_CLIENT_VM%
```

Updating WebLogic Server 7.0 to 7.0 Service Pack 2

If you installed the stand-alone WebLogic Server 7.0 and did not upgrade to the WebLogic Platform Edition you cannot use Smart Update to install WebLogic Server 7.0 Service Pack 1. Instead you can use a package upgrade installer that you can download from the BEA Customer Support Web site at http://support.bea.com. For instructions, see Installing Service Packs and Rolling Patches with a Downloadable Installer in *Installation Guide*.

To see which version of WebLogic Server you have installed, open *BEA_HOME*\logs\log.txt (*BEA_HOME*/logs/log.txt on UNIX) in a text editor and look at the last entry in the log file. Each line in the log file represents an installation event for the current BEA home—an installation or an uninstallation. If the log entry indicates that the version installed is WebLogic Platform Edition 7.0 SP0 or earlier, you cannot use Smart Update. If the log entry indicates that the version installed is WebLogic Platform Edition (7.0.0.1) or later, you *can* use Smart Update to install WebLogic Server Service Pack 2.

WebLogic Server and WebLogic Platform Edition use the same installation framework. Therefore, installation log entries use the term *WebLogic Platform* to indicate an installation of either WebLogic Server or WebLogic Platform Edition.

Performance Enhancements

WebLogic Server 7.0 SP2 introduces important performance-related enhancements:

- Adjustment of out-of-the-box (OOTB) parameters to realize greater performance when you execute applications using the default configuration. These adjustments do not result in unwanted side-effects.

- The `PreparedStatementCache` functionality is now turned on by default, which reduces both network roundtrips and preparation work in the database. See "Increasing Performance with the Prepared Statement Cache" for more details about the Prepared Statement Cache, including usage restrictions.

- MSSQLServer Pessimistic Concurrency enables WebLogic Server to use advanced functionality in Microsoft SQLServer that was only previously available in Oracle. This functionality is required for certain benchmarking efforts and may have a positive effect on performance.

- Value Change Check is an additional check to eliminate unnecessary writes to the database for EJB attributes. The largest performance benefit is realized with long-valued columns (large chars/varchars, and Blobs/Clobs).

New and Changed Security Documentation

A new manual, *Securing WebLogic Resources* at
http://e-docs.bea.com/wls/docs70/secwlres/index.html is available for this release of WebLogic Server.
In addition, *Introduction to WebLogic Security* at
http://e-docs.bea.com/wls/docs70/secintro/index.html, *Managing WebLogic Security* at
http://e-docs.bea.com/wls/docs70/secmanage/index.html, and *Programming WebLogic Security* at
http://e-docs.bea.com/wls/docs70/security/index.html have been significantly updated.

Web Services

WebLogic Server 7.0 SP2 includes support for the following new Web services features:

- Memory-use optimization

  Optimize memory use during Webservice invocations by caching the Web Services Descriptor Language (WSDL). To take advantage of this feature, use the new `weblogic.webservice.cachewsdl` flag when starting the server.

- Portable stubs

  If you use the Web services client JAR files (those distributed with the product and the Web service-specific file generated by the `clientgen` Ant task) as part of an application that runs in WebLogic Server, the Java classes in the JAR file might collide with the classes of WebLogic Server itself. This problem is more apparent if the WebLogic Server instance in which the client

JAR file is deployed is a different version from the version in which the client JAR file was generated. To solve this problem, use portable stubs.

Note: You need to use portable stubs only if your client application is deployed and running on WebLogic Server. If your client application is standalone, you do not need to use portable stubs.

To enable your client application to use portable stubs, follow the instructions in Creating and Using Portable Stubs in *Programming WebLogic Web Services*.

- `Autotype` and `servicegen` ant tasks handle additional non-built-in data types.

  The autotype and servicegen Ant tasks now handle additional non-built-in datatypes, including the following:

  - `<xsd:complexType>` with simple content
  - `java.util.Stack`
  - JAX-RPC-style enumeration class

  For details, see Non-Built-In Data Types Supported by the servicegen and autotype Ant Tasks.

Optimized JMS Topic Subscriber Message Selectors

For a certain class of applications, WebLogic JMS can optimize topic subscriber message selectors significantly by indexing them. These applications have messages that contain one or more unique identifiers and thousands of subscribers that filter based on these identifiers.

A typical example is an instant messaging application where each subscriber corresponds to a different user, and each message contains a list of one or more target users. See Indexing Topic Subscriber Message Selectors To Optimize Performance in *Programming WebLogic JMS*.

Extra EJB Compiler Options at the Server Level

You can now specify extra EJB Compiler options in the Administration Console at the server level—under Server -> Configuration—in addition to at the EJB module level—under EJB -> Configuration ->Advanced options.

Note: The EJB-level settings take precedence. If the EJB-level options are set to null on the `EJBComponentMbean`, WebLogic Server applies the value specified at the server level, on the `ServerMBean`.

New Features for Installing WebLogic Server as a Windows Service

The WebLogic Server utility that sets up WebLogic Servers to run as Windows services has been modified:

- It provides a new option that enables you to gracefully shut down a WebLogic Server Windows service from the Windows Control Panel. The new option, `-stopclass`, is documented in the Administration Guide.

- The behavior of the `-delay` option has been modified. In previous releases, this option specified the number of milliseconds before the Windows Service Control Manager (SCM) would invoke a Windows service that has specified a dependency on another service. In this release, the `-delay` option specifies the number of milliseconds the Windows SCM waits before it changes the status of the associated Windows service from STARTING to STARTED. For more information, refer to the Administration Guide.

New Default Version for the Oracle Thin Driver

The default Oracle JDBC Thin driver has been changed to the 9.2.0 version. In releases of WebLogic Server 7.0 prior to the Service Pack 2 release, the 8.1.7 version of the Oracle Thin driver was the default version.

For more information about using different versions of the Oracle Thin driver, see Changing or Updating the Oracle Thin Driver in *Programming WebLogic JDBC*.

What's New in WebLogic Server 7.0 SP1

The following sections contain important notes for BEA WebLogic Server 7.0 SP1.

JMS

Service Pack 1 offers new JMS features.

Full Compliance with the JMS Specification 1.0.2b

WebLogic JMS is now fully compliant with Sun Microsystems' JMS Specification version 1.0.2b. As a result, message property names must now strictly conform to the message selector syntax specifications for Java identifiers, as defined in the javax.jms.Message Javadoc.

Synchronous Write Policy for JMS File Stores

In WebLogic Server 7.0 Service Pack 1, WebLogic JMS file stores guarantee up-to-the-message integrity by using synchronous writes by default. Disabling synchronous writes improves file store performance, often dramatically, but at the expense of possibly losing sent messages or generating duplicate received messages (even if the messages are transactional) in the event of an operating system crash or a hardware failure.

See Configuring a Synchronous Write Policy for JMS File Stores in the Administration Guide.

Servlets

Beginning with WebLogic Server 7.0 Service Pack 1, a servlet's initialization is invoked as the `<init-as>` user, specified in `weblogic.xml`. If not specified in `weblogic.xml`, then it is invoked as the `<run-as>` user, specified in `web.xml` If not specified here either, then it is invoked as the `<anonymous>` user.

As a result of this change, the `<init-as>` and `<destroy-as>` elements were added to the `weblogic.xml` deployment descriptor. For more information, see "weblogic.xml Deployment Descriptor Elements" in *Developing WebLogic Server Web Applications*.

Web Services

Service Pack 1 of WebLogic Web services includes support for implementing document-oriented Web service operations. As described in the WSDL 1.1 specification, document-oriented Web service operations are those in which the SOAP messages contain documents, as opposed to RPC-oriented Web services whose SOAP messages contain parameters and return values. For details, see *RPC-Oriented or Document-Oriented Web Services* at http://e-docs.bea.com/wls/docs70/webserv/design.html#develop001.

Service Pack 1 of WebLogic Web Services also includes the following new Ant tasks:

- `autotype`

  Generates the serialization class, Java representation, XML Schema representation, and data type mapping information for non-built-in data types used as parameters or return values to a WebLogic Web service.

- `source2wsdd`

  Generates a `web-services.xml` deployment descriptor file from the Java source file for a Java class-implemented WebLogic Web service.

- `wsdl2Service`

Generates the components of a WebLogic Web service from a WSDL file. The components include the `web-services.xml` deployment descriptor file and a Java source file that you can use as a starting point to implement the Web service.

- `wspackage`

  Packages the components of a WebLogic Web service into a deployable EAR file.

In addition, the `servicegen` Ant task has the following new attributes used to assemble a JMS-implemented Web service:

- `JMSDestination`
- `JMSDestinationType`
- `JMSAction`
- `JMSConnectionFactory`
- `JMSOperationName`
- `JMSMessageType`

For reference information about these Ant tasks, see *Web Service Ant Tasks and Command-Line Utilities* at http://e-docs.bea.com/wls/docs70/webserv/anttasks.html.

Note: The `wsdl2Service` Ant task is a preliminary implementation and has not been fully tested in this release.

WebLogic jDriver for Oracle 9.2.0

With the release of WebLogic Server 7.0 Service Pack 1, WebLogic jDriver for Oracle version 9.2.0 and Oracle Thin Driver version 9.2.0 are certified for use with WebLogic Server 7.0. You can use these drivers to connect to an Oracle 9.2.0 database and create database connections in a JDBC connection pool.

For more information, see the following related documents:

- *Configuring WebLogic jDriver for Oracle* at
  http://e-docs.bea.com/wls/docs70/oracle/install_jdbc.html

- *Using Third-Party JDBC Drivers with WebLogic Server* at
  http://e-docs.bea.com/wls/docs70/jdbc/thirdparty.html

- WebLogic Server Certifications at
  http://e-docs.bea.com/platform/suppconfigs/index.html

New Security-Related Element for EJBs

The 7.0 SP1 release introduces a new `weblogic-ejb-jar.xml` element: `global-role`. This element indicates that a particular security role is defined "globally" in a security realm. See *Programming WebLogic Enterprise JavaBeans* for details.

WebLogic Server 7.0 Platform Edition and WebLogic Server 7.0 Standalone

WebLogic Server 7.0 Platform Edition, released on June 28, 2002, incorporated two new features that are also included in WebLogic Server 7.0 SP1. If you downloaded WebLogic Server before that date you have the WebLogic Server 7.0 standalone version. WebLogic Server 7.0 Platform Edition differs from WebLogic Server 7.0 stand-alone as follows:

- WebLogic Workshop, a powerful integrated development environment, was added to WebLogic Server 7.0 Platform Edition.

- Also, in the Platform Edition, WebLogic Server no longer requires the download of a separate JAR file to enable Web Services.

WebLogic Server 7.0 SP1 incorporates these features.

Updating WebLogic Server 7.0 to 7.0 Service Pack 1

If you installed the stand-alone WebLogic Server 7.0 and did not upgrade to the WebLogic Platform Edition, released June 28, 2002, you cannot use Smart Update to install WebLogic Server 7.0 Service Pack 1. Instead you can use a package upgrade installer that you can download from the BEA Customer Support Web site at http://support.bea.com. For instructions, see Installing Service Packs and Rolling Patches with a Downloadable Installer in *Installation Guide*.

To see which version of WebLogic Server you have installed, open *BEA_HOME*\logs\log.txt (*BEA_HOME*/logs/log.txt on UNIX) in a text editor and look at the last entry in the log file. Each line in the log file represents an installation event for the current BEA home—an installation or an uninstallation. If the log entry indicates that the version installed is WebLogic Platform Edition 7.0 SP0 or earlier, you cannot use Smart Update. If the log entry indicates that the version installed is WebLogic Platform Edition (7.0.0.1) or later, you *can* use Smart Update to install WebLogic Server Service Pack 1.

Note: WebLogic Server and WebLogic Platform Edition use the same installation framework. Therefore, installation log entries use the term *WebLogic Platform* to indicate an installation of either WebLogic Server or WebLogic Platform Edition.

Migrating Domains Created Using the Configuration Wizard

The Configuration Wizard (introduced in WebLogic Platform 7.0) allows you to create new domains quickly and easily. If you created domains using the Configuration Wizard in WebLogic Platform 7.0, you need to migrate those domains for use with WebLogic Platform 7.0 Service Pack 1.

For most domains, migration is a three-step process:

1. Upgrade the product JAR files in the domain directory. A migration script is provided for this purpose. See instructions in "Step 1: Upgrade Product JAR Files" on page 1-20.

   Note: You can also revert a domain to its pre-migration state.

2. Update the domain to support Service Pack 1 changes. Depending on the domain template used to generate the domain, you may need to add or modify existing scripts or files. See instructions in "Step 2: Update Domain to Support Service Pack 1 Changes" on page 1-21.

3. If you installed WebLogic Platform 7.0 Service Pack 1 into a new directory, separate from the WebLogic Platform 7.0 installation, update the domain startup scripts and configuration files to reference the new *BEA_HOME* directory location. See instructions in "Step 3: Update Startup Scripts and Configuration Files to Reference New BEA_HOME Directory Location (Non-Upgrades Only)" on page 1-25.

   Note: If you *upgraded* your existing WebLogic Platform 7.0 installation, you can skip this step.

You will need to repeat this process for *each* domain that you want to migrate.

Note: This section describes how to migrate domains specific to WebLogic Server. For information about migrating other WebLogic Platform domains, see "Migrating Domains Created Using the Configuration Wizard" in the *WebLogic Platform 7.0 Service Pack 2 Release Notes* at the following URL:

   http://e-docs.bea.com/platform/docs70/relnotes/relnotes.html#migration

Step 1: Upgrade Product JAR Files

To upgrade product JAR files for a 7.0 domain that you generated using the Configuration Wizard to Service Pack 1, navigate to the *BEA_HOME*\weblogic700\server\bin directory and execute one of the following commands:

Windows: migrate.cmd *domain mode*
UNIX: migrate.sh *domain mode*

Note: You will be prompted to press any key to start processing.

The following table defines the command-line arguments.

Table 1-1 Command-line Arguments for Upgrading 7.0 Product JAR files to 7.0 SP1

Argument	Description
domain	Full pathname of the domain directory.
mode	Migration mode. The mode can be set to one of the following values: • `upgrade`: Upgrade the product JAR files in the domain directory, as required. The original product JAR files are saved as `*.jar.orig`. If the timestamp of an existing product JAR file is more recent than the timestamp on the corresponding SP1 installation product JAR file, the file is skipped. This is the default mode. • `revert`: Reverts a domain that was migrated earlier using the backup files (`*.jar.orig`) generated. If no `*.jar.orig` files exist, the command is ignored.

For example, the following command upgrades a domain called `mydomain` located in the default user projects directory (*BEA_HOME*\user_projects):

Windows: `migrate.cmd c:\bea\user_projects\mydomain upgrade`
UNIX: `migrate.sh c:/bea/user_projects/mydomain upgrade`

The following command reverts the changes made to `mydomain` during the migration:

Windows: `migrate.cmd c:\bea\user_projects\mydomain revert`
UNIX: `migrate.sh c:/bea/user_projects/mydomain revert`

Step 2: Update Domain to Support Service Pack 1 Changes

Depending on the domain template used to generate the domain, you may need to add or modify existing scripts or files to support WebLogic Platform 7.0 Service Pack 1 changes. Refer to the appropriate section below, based on the domain template used to generate the domain, for additional migration steps:

- WebLogic Workshop Domain

- WLS Domain

- WLS Examples

- WLS Petstore

Note: Before adding or modifying any files, as described in the following sections, it is recommended that you backup the original files.

WebLogic Workshop Domain

For a domain that is based on the WebLogic Workshop template, perform the following steps:

1. Modify the startWebLogic.cmd (Windows) or startWebLogic.sh (UNIX) command to reflect the appropriate PointBase version (183 versus 172) in the JAR filenames defined in the CLASSPATH. The files for both commands are located in the following directory, by default:

   BEA_HOME\user_projects*domain*

   The following sample excerpt from the startWebLogic.cmd script (Windows) shows the required updates in **bold**:

   Before:

   ```
   set PB_CLASSPATH=
   %POINTBASEDIR%\eval\pointbase\lib\pbserver42ECF172.jar;
   %POINTBASEDIR%\eval\pointbase\lib\pbclient42ECF172.jar
   ```

After:

```
set PB_CLASSPATH=.\;
%POINTBASEDIR%\eval\pointbase\lib\pbserver42ECF183.jar;
%POINTBASEDIR%\eval\pointbase\lib\pbclient42ECF183.jar
```

2. Copy the following files from the *BEA_HOME*\weblogic700\samples\workshop directory to the *BEA_HOME*\user_projects*domain* directory of your WebLogic Workshop domain. Be careful not to overwrite any files that may have been created using one of these filenames.

```
setWorkshopEnv.cmd
setWorkshopEnv.sh
startPointBaseConsole.cmd
startPointBaseConsole.sh
URLs.dat
```

WLS Domain

For a domain that is based on the WLS Domain template, you do not need to add or modify existing scripts or files.

WLS Examples

For a domain that is based on the WLS Examples domain template, perform the following steps:

1. Modify the CLASSPATH definition in the startExamplesServer.cmd (Windows) or startExamplesServer.sh (UNIX) command to:

    – Reflect the appropriate PointBase version (183 versus 172) in the JAR filenames.

    – Add the *BEA_HOME*\server\lib\webservices.jar file.

    The files for both commands are located in the following directory, by default:

    BEA_HOME\user_projects*domain*

    The following sample excerpt from the startExamplesServer.cmd script (Windows) shows the required updates in **bold**:

Before:

```
set CLASSPATH=
c:\bea\jdk131_03\lib\tools.jar;%POINTBASE_HOME%\lib\
pbserver42ECF172.jar;%POINTBASE_HOME%\lib\
pbclient42ECF172.jar;%CLIENT_CLASSES%;%SERVER_CLASSES%;
%COMMON_CLASSES%;%CLIENT_CLASSES%\utils_common.jar
```

After:

```
set CLASSPATH=
c:\bea\jdk131_03\lib\tools.jar;%POINTBASE_HOME%\lib\
pbserver42ECF183.jar;%POINTBASE_HOME%\lib\
pbclient42ECF183.jar;%CLIENT_CLASSES%;%SERVER_CLASSES%;
%COMMON_CLASSES%;%CLIENT_CLASSES%\utils_common.jar;
c:\bea\weblogic700\server\lib\webservices.jar
```

2. Copy the `Webservices_trader.ear` file from the
   `BEA_HOME`\samples\server\config\examples\applications directory to the
   `BEA_HOME`\user_projects\WLSExampleDomain\applications directory of your Web
   applications. Be careful not to overwrite any files that you have customized.

WLS Petstore

For a domain that is based on the WLS Petstore domain template, modify the `startPetStore.cmd`
(Windows) or `startPetStore.sh` (UNIX) command to reflect the appropriate PointBase version
(183 versus 172) in the JAR filenames defined in the CLASSPATH.

The files for both commands are located in the following directory, by default:

`BEA_HOME`\user_projects*domain*

The following sample excerpt from the `startPetStore.cmd` script (Windows) shows the required
updates in **bold**:

Before:

```
set
CLASSPATH=%JAVA_HOME%\lib\tools.jar;%POINTBASE_HOME%\lib\pbserver42ECF1
72.jar;%POINTBASE_HOME%\lib\pbclient42ECF172.jar;%SERVER_CLASSES%;%COMM
ON_CLASSES%
```

After:

```
set
CLASSPATH=%JAVA_HOME%\lib\tools.jar;%POINTBASE_HOME%\lib\pbserver42ECF1
83.jar;%POINTBASE_HOME%\lib\pbclient42ECF183.jar;%SERVER_CLASSES%;%COMM
ON_CLASSES%
```

Step 3: Update Startup Scripts and Configuration Files to Reference New BEA_HOME Directory Location (Non-Upgrades Only)

Note: This step is only required if you installed WebLogic Platform 7.0 Service Pack 1 into a new directory that is separate from the WebLogic Platform 7.0 installation. If you *upgraded* your existing WebLogic Platform 7.0 installation, you can skip this step.

The domain startup scripts (such as, `startWebLogic`) and configuration files (such as `config.xml`) define the full pathnames to files within the *BEA_HOME* directory. You need to search for and update these full pathnames to reference the new *BEA_HOME* directory location. In addition, you must update any custom scripts, such as build scripts, that define full pathnames to the files within the *BEA_HOME* directory to reflect the new *BEA_HOME* location.

Many startup scripts set environment variables in your current shell, including variables that reference your *BEA_HOME* directory. After updating the *BEA_HOME* references in script files, you should open a new shell to ensure that the latest environment settings are used.

What's New in WebLogic Server 7.0

BEA WebLogic Server changed significantly in the 7.0 release. This section details major differences between WebLogic Server 7.0 and earlier versions. Later sections describe additional changes in service pack releases.

Web Services

New features provide additional flexibility, ensure interoperability with other key Web Services vendors, and allow J2EE programmers to easily expose J2EE components as Web Services. The programming model shields the user from the complexities of SOAP and WSDL while providing optional extensibility. The new features include:

- Implementation of version 1.0 of the *Java API for XML Based RPC* (JAX-RPC)

- Enhanced Ant tasks to automate many of the Web service programming tasks

- Support for user-defined data types

- Message handlers for intercepting SOAP message requests and responses

- Point-to-point security

- Support for J2ME clients

- UDDI registry and client API

For more information about WebLogic Web services, see *Programming WebLogic Web Services* at
`http://e-docs.bea.com/wls/docs70/webserv/index.html`.

Security Infrastructure

The completely redesigned WebLogic Security Service offers a modular design that exposes a set of fully implemented SSPIs (Security Service Provider Interfaces) for authentication, authorization, auditing, role mapping, credential mapping, and keystore (PKI) management. Modules from third-party security vendors can plug right into the WebLogic Server Framework, and third-party administration tools can be integrated into the Administration Console. A new role-based authorization module can be applied to all J2EE and non-J2EE resources, and an embedded security policy engine makes it easy to create prose-based rules for dynamically assigning roles and access privileges. See the *WebLogic Security Documentation* at
`http://e-docs.bea.com/wls/docs70/security.html`.

Developer Tools

WebLogic Builder prepares Java files for quick deployment to the application server. WebLogic Builder is a graphical tool for assembling a J2EE application module, creating and editing its deployment descriptors, and deploying it to WebLogic Server. To run the tool, start the server, set your environment in a shell and type: `java weblogic.marathon.Main`. For more information, see the *WebLogic Builder Online Help* at
`http://e-docs.bea.com/wls/docs70/wlbuilder/index.html`.

EJBGen is an EJB generation tool that uses javadoc comments to generate, from a single bean source file, deployment descriptor files and the EJB home, local, and remote interfaces. For more information, see *Programming WebLogic Enterprise JavaBeans* at
`http://e-docs.bea.com/wls/docs70/ejb/index.html`.

The `weblogic.Deployer` utility replaces the earlier `weblogic.deploy` utility. Use `weblogic.Deployer` to deploy a new application; redeploy an application; redeploy part of an application; deactivate an application; undeploy an application; cancel a deployment task; or list all deployment tasks, all from a simple command line interface.

WebLogic Server 7.0.0.1 and later versions also include BEA WebLogic Workshop™, an integrated development framework that empowers all application developers — not just J2EE experts — to rapidly create, test, and deploy enterprise-class Web Service applications on the BEA WebLogic Enterprise Platform™. BEA WebLogic Workshop provides a unified development platform that enables developers to easily build and connect components, data, and application business logic, while insulating them from the complexities of J2EE. For the first time, application and enterprise

developers can work together on the same platform in the same language, dramatically increasing productivity across IT organizations.

To get the details on these tools and more see our WebLogic Server *Tools and Utilities Guide,* at `http://e-docs.bea.com/wls/docs70/toolstable/index.html`.

System Administration Enhancements

The following administrative improvements and new features were added in WebLogic Server 7.0:

- Improved management of the server lifecycle. As servers start up and shut down they cycle through several operational states which can be monitored and controlled remotely. These states include a hot standby mode that allows you to keep a server ready to be brought online immediately when needed.

- Managed Server independence. WebLogic Server instances now have the ability to start even when the instance cannot contact the Administration Server.

- Improved application deployment. Application deployment now uses a two-phase deployment model. New deployment screens in the Administration Console are provided as well as a new command-line tool.

- Migratable Services. Certain services deployed on a single server instance in a cluster, such as JMS and JTA, can be migrated to other servers in the event of server failure without interrupting client requests.

- Network Channels. Network channels enable you to configure additional ports with one or more WebLogic Server instances or clusters. Support for using multiple Network Interface Cards is also provided.

- Domain health monitoring.You can now monitor the health of servers and applications in a domain. A cluster can automatically respond to failed servers by directing client requests to other cluster members or rebooting a server.

    Servers can also monitor their own health. Applications can make use of the Node Manager to monitor servers and take corrective action if a server fails.

- Improved help available for the Administration Console. Online, context-sensitive help for every attribute and screen is now available from within the Administration Console.

Caching

WebLogic Server 6.0 introduced JSP Cache Tags to cache specific portions of JSPs in memory without unnecessary trips to the backend; WebLogic Server 7.0 CacheFilters enable you to configure caching

easily for entire pages, URLs, and file types. Without requiring any code changes to the application, administrators can turn on caching and see immediate performance and scalability improvements.

Clustering

You can now set up clusters without multihoming your machine. In earlier releases, a cluster required a unique listen address for each server while each server listened on the same port. If you wanted to create a cluster on a single computer, you were required to set up a multihomed environment, with multiple IP addresses on a single computer. Now you can assign different listen port numbers to the different servers in the cluster, resulting in a clustered environment on a single machine with one listen address. For more information, see *Using WebLogic Server Clusters* at
http://e-docs.bea.com/wls/docs70/cluster/index.html.

EJB 2.0

New Enterprise JavaBeans (EJB) 2.0 features include dynamic query support for executing queries within application code; multiple DBMS table mapping with WebLogic Server container-managed persistence services; new EJB WebLogic QL support for subqueries, aggregate functions, queries that return result sets, case-insensitive searching and SELECT FOR UPDATE with NO WAIT statements; improved concurrency and caching; EJB link support; and support for bulk insert update. For more information about all new EJB features, see *Programming WebLogic Enterprise JavaBeans* at
http://e-docs.bea.com/wls/docs70/ejb/index.html.

Message-Driven Beans and Non-BEA JMS Providers

Beginning with WebLogic Server 7.0, applications with transactional MDBs can achieve exactly-once semantics with a non-BEA JMS provider for messages processed by an MDB. In addition, WebLogic Server will use XA to automatically enlist the non-BEA JMS provider in a transaction. For details, see "Configuring Message-Driven Beans for non-BEA JMS Providers" at http://e-docs.bea.com/wls/docs70/ejb/message_beans.html#MDB_with_foreign_JMS.

Directory Structure

BEA changed the directory structure and the location of applications built on WebLogic Server 7.0. This new directory structure provides added flexibility and promotes best practices for application development. See the BEA Home Directory section in the *WebLogic Server Installation Guide* at
http://e-docs.bea.com/wls/docs70/install/instpre.html for more information.

Configuration Wizard

Domain and cluster management are greatly simplified by a Configuration Wizard that lets you generate pre-configured domains in any location. A domain is an interrelated set of WebLogic Server resources that share configuration files and are managed as a unit that includes WebLogic Server instances, WebLogic Server clusters, and applications. Based on user queries, the Configuration Wizard generates a domain, server, and enterprise application with the appropriate components pre-configured and assets included. You can run the Domain Configuration Wizard during a custom installation and anytime thereafter. The Configuration Wizard is especially designed to make setting up clusters easy. The scripts for starting the Configuration Wizard are located in your *WL_HOME*\common\bin directory.

Installation

The following installation enhancements are new to this release of WebLogic Server. For more detailed information, see the *Installation Guide* at
http://e-docs.bea.com/wls/docs70/install/index.html.

Installation Options

When installing WebLogic Server 7.0, you can choose between typical and custom installation options. The typical installation includes the most common options for quickly installing WebLogic Server. With the custom installation, you can choose installation options and configure a domain during the installation.

Smart Updates

WebLogic Server 7.0 includes a new installer program that you can also use to find and install service packs and product updates.

J2EE Connectors

J2EE Connectors define a standard method of integrating WebLogic Server applications with existing applications. WebLogic Server 7.0 implements the finalized version of the Connectors 1.0 specification. This release also includes enhancements to connection and security management for JCA adapters.

The following J2EE Connector Architecture enhancements are new to this release of WebLogic Server. For more detailed information, see *Programming the WebLogic J2EE Connectors* at
http://e-docs.bea.com/wls/docs70/jconnector/index.html.

JNDI Implementation

This release supports a referencable JNDI implementation. BEA has provided united testing to confirm that previous related problems have been resolved.

Secure Password Credential Storage

This release provides a standard method for resource adapter deployers to plug in their specified authorization/authentication mechanism through secure password credential storage. This WebLogic Server storage mechanism has replaced the Security Principal Mapping mechanism provided with the `weblogic-ra.xml` deployment descriptor within the resource adapter archive.

This new storage mechanism is used to map initiating principals (such as WebLogic Server username and password combinations) to resource principals (EIS user name and password combinations).

Connection Leak Detection

In the past release, the connection leak detection mechanism was based upon a timer that started when a connection was created and triggered when the connection exceeded its usage duration. WebLogic Server now provides new mechanisms for preventing this scenario, leveraging a garbage collector and providing an idle timer for tracking the usage of connection objects.

Dynamic Pool Modification

This release provides dynamic runtime changes to the various connection pool parameters that are configurable at runtime. When you make changes to these parameters using the Console Deployment Descriptor Editor, the changes take place immediately. You do not need to reboot WebLogic Server and redeploy the resource adapter.

Security Policy Processing of a ra.xml Specification

BEA WebLogic Server J2EE Connector Architecture provides a set of security permissions for execution of a resource adapter in a managed runtime environment. WebLogic Server also grants a resource adapter explicit permissions to access system resources.

jCOM

WebLogic jCOM is a software bridge that allows bi-directional access between Java and J2EE objects deployed in WebLogic Server, and Microsoft ActiveX components available within Microsoft Office family of products, VisualBasic and C++ objects, and other COM/DCOM-compliant environments.

WebLogic jCOM allows Microsoft COM clients to access objects in WebLogic Server as though they were COM components and applications within WebLogic Server to access COM components as though they were Java objects. In both cases, WebLogic jCOM makes the differences between the object types transparent: to a COM client, WebLogic Server objects appear to be COM objects and to a WebLogic Server application, COM components appear to be Java objects.

JDBC

The following are new or improved features of WebLogic JDBC.

Support for ARRAYs, STRUCTs, and REFs

WebLogic Server 7.0 includes support for ARRAYs, STRUCTs, and REFs, including the standard JDBC methods and some Oracle extensions. To use ARRAYs, STRUCTs, or REFs, you must use the Oracle Thin Driver. See Oracle Thin Driver Extensions in *Programming WebLogic JDBC* at
`http://e-docs.bea.com/wls/docs70/jdbc/index.html` for more information.

JDBC Connection Pool Administration MBean

WebLogic Server provides the `JDBCConnectionPool` administration MBean as part of the WebLogic Server management architecture (JMX). You can use the `JDBCConnectionPool` MBean to dynamically create and configure a connection pool from within a Java application. See Creating a Connection Pool Dynamically in *Programming WebLogic JDBC* at
`http://e-docs.bea.com/wls/docs70/jdbc/index.html` for more information.

Third-Party JDBC Drivers Shipped with WebLogic Server

WebLogic Server includes several versions of the Oracle Thin Driver and the Sybase jConnect JDBC driver stored in different folders in the directory structure. You can select which version of the JDBC driver you want to use. You can also easily update a driver with a version from the DBMS vendor. See Overview of Third-Party JDBC Drivers in *Programming WebLogic JDBC* at
`http://e-docs.bea.com/wls/docs70/jdbc/index.html` for more information.

Oracle 9i Support

WebLogic Server now supports Oracle 9i. To connect to an Oracle 9i database, you must use a JDBC driver that supports Oracle 9i, such as the WebLogic jDriver for Oracle or the Oracle Thin Driver version 9.0.1 or later. For more information, see *Installing and Using WebLogic jDriver for Oracle* and *Programming WebLogic JDBC* at
`http://e-docs.bea.com/wls/docs70/jdbc/index.html`.

JMS

The following are new features of WebLogic JMS.

Enhanced Availability

WebLogic JMS takes advantage of the migration framework implemented in the WebLogic Server core for clustered environments. This allows WebLogic JMS to respond properly to migration requests and bring a JMS server online and offline in an orderly fashion. This includes both scheduled migrations as well as migrations in response to a WebLogic Server failure.

For more information, see "Managing JMS" in *Programming WebLogic JMS* at `http://e-docs.bea.com/wls/docs70/jms/config.html`.

Distributed Destinations within WebLogic Clusters

The highly available implementation of WebLogic JMS offers a level of service continuity in the event of a single server failure within a cluster by enabling you to configure multiple physical destinations as members of a single distributed destination set. Specifically, an administrator can configure multiple instances of a given destination (queue or topic) within a cluster. If one instance within the cluster fails, then other instances of the same distributed destination will be able to provide service to JMS producers and consumers.

For more information, see "Developing a WebLogic JMS Application" in *Programming WebLogic JMS* at `http://e-docs.bea.com/wls/docs70/jms/implement.html` and "Managing JMS" in the *Administration Guide* at `http://e-docs.bea.com/wls/docs70/adminguide/jms.html`.

Flow Control

Using the Flow Control feature, you can enable a JMS server or destination to slow down message producers when it determines that it is becoming overloaded. Specifically, when a JMS server or destination exceeds its specified bytes or messages thresholds, it instructs producers to limit their message flow.

For more information, see "Managing JMS" in the *Administration Guide* at `http://e-docs.bea.com/wls/docs70/adminguide/jms.html`.

WebLogic Messaging Bridge

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

For more information, see "Configuring a Messaging Bridge" in the *Administration Guide* at
`http://e-docs.bea.com/wls/docs70/adminguide/msgbridge.html`.

Message Paging

The 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.

For more information, see "Managing JMS" in the *Administration Guide* at
`http://e-docs.bea.com/wls/docs70/adminguide/jms.html`.

Server Affinity With Destination Lookup

The `createTopic()` and `createQueue()` methods now allow a
"`JMS_Server_Name./Destination_Name`" syntax to indicate server affinity when you are looking up destinations. When a destination is locally deployed in the same JVM as the connection factory, the connection factory only returns names matching local destinations. If the name is not on the local JVM an exception is thrown, even though the same name might be deployed on a different JVM.

An application can use this convention to avoid hard-coding the server name when using the `createTopic` and `createQueue` methods so that the code can be re-used as is on other JMS servers. For more information, see Developing a WebLogic JMS Application in *Programming WebLogic JMS* at http://e-docs.bea.com/wls/docs70/jms/implement.html.

BEA Jolt Client 8.0.1

BEA Jolt Client 8.0.1 is a Java-based client API that manages requests to Tuxedo services via a Jolt Service Listener (JSL) running on the Tuxedo server. The Jolt API is embedded within the WebLogic API and is accessible from a servlet or any other BEA WebLogic application.

The Jolt Java Client 8.0.1 is available from the BEA Product Download Center at http://commerce.bea.com/downloads/products.jsp. Follow the link to BEA WebLogic Server 7.0 and select the Jolt Java Client 8.0.1 from the Modules for WebLogic Server list. A Jolt license is required.

Java Transaction API (JTA)

The following Java Transaction API enhancement is new to this release of WebLogic Server, For more detailed information see, *Programming WebLogic JTA* at

`http://e-docs.bea.com/wls/docs70/jta/index.html.`

Transaction Recovery Service Migration

In WebLogic Server 7.0, you can migrate the Transaction Recovery Service from one server in a cluster to another server in the same cluster, which provides highly-available transaction recovery. The Transaction Recovery Service completes incomplete transactions for a failed server. See Transaction Recovery After a Server Fails in the *WebLogic Server Administration Guide* at

`http://e-docs.bea.com/wls/docs70/adminguide.`

Network Resources

WebLogic Server 7.0 introduces two new types of configurable network resources that you can administer using either the Administration Console or WebLogic Server MBeans. These resources are network channels and Network Access Points (NAPs).

A network channel defines the basic attributes of a network connection to WebLogic Server. You configure channels as distinct entities in the Administration Console, and then add one or more channels to servers in a domain. Using a single channel with multiple servers can simplify network configuration for a WebLogic Server domain, because changing the channel configuration automatically changes the connection attributes of all servers that use the channel. You can also create and assign multiple channels to a single server. Using multiple channels helps you segment network traffic by protocol, listen ports, or any other channel configuration property. For example, you can use two channels with a single server to tailor the default connection properties for secure vs. non-secure traffic. For more information, see *Configuring Network Resources* at

`http://e-docs.bea.com/wls/docs70/docs70/admin_domain/network.html.`

A Network Access Point (NAP) is a configurable resource that represents the actual port numbers and, optionally, the specific IP address that a network channel uses. Although network channels alone can help you manage shared connection properties in a domain, channels are more commonly used in conjunction with NAPs to segment network traffic among multiple NICs and port numbers. NAPs also override certain channel attributes to customize a WebLogic Server's network configuration. For more information, see *Configuring Network Resources* at

`http://e-docs.bea.com/wls/docs70/docs70/admin_domain/network.html.`

RMI-IIOP

The WebLogic implementation of RMI-IIOP now includes the following features, as well as numerous performance enhancements:

- Support for Object-Transaction-Services 1.2 (OTS 1.2) which adds transactional support to RMI-IIOP. Can be used to invoke remote objects hosted in foreign application servers.

- CSIv2 (Common Secure Interoperability, version 2) provides client authentication, delegation, and privilege functionality; also works with foreign application servers.

- A WebLogic IIOP Client that is fully clusterable. Simply use the following command line parameter to enable support for the WebLogic IIOP Client: `-D weblogic.system.iiop.enableClient=true`.

- Hot code generation of IIOP stubs. There is no longer any need to use the `-iiop` flag with `ejbc` or `rmic`.

- Support for the entire CosNaming API

Updated Security Configuration Procedure for Web Applications and EJBs

The procedure for protecting Web applications and EJBs has been improved in this release of WebLogic Server. For a complete description, see *Securing WebLogic Resources*.

WebLogic Tuxedo Connector

WebLogic Tuxedo Connector now includes the following features:

- WebLogic Tuxedo Connector is a fully integrated WebLogic Server Service and no longer uses a separate XML configuration file

- Codeset translation is provided using the WebLogic Server weblogic.wtc.encoding property

- View Buffer support

- Ability to work within a clustered environment

WebLogic XML

WebLogic Server 7.0 includes the following new XML features:

- XML Streaming API: The WebLogic XML Streaming API provides an easy and intuitive way to parse and generate XML documents. It is similar to the SAX API, but enables a procedural, stream-based handling of XML documents rather than requiring you to write SAX event handlers, which can get complicated when you work with complex XML documents. In other words, the streaming API gives you more control over parsing than the SAX API.

- XML Application Scoping: Application scoping refers to configuring resources for a particular enterprise application rather than for an entire WebLogic Server configuration. In the case of XML, these resources include parser, transformer, external entity, and external entity cache configuration. The main advantage of application scoping is that it isolates the resources for a given application to the application itself. Using application scoping, you can configure different parsers for different applications, store the DTDs for an application within the EAR file or exploded enterprise directory, and so on.

For more information about these new features, see *Programming WebLogic XML* at
`http://e-docs.bea.com/wls/docs70/xml/index.html`.

Documentation and Examples

The following items include enhancements to the documentation and the samples provided with WebLogic Server 7.0.

Documentation

E-docs at `http://edocs/index.html` is the source for WebLogic Server information. The documentation covers all the newest features and changes in WebLogic Server 7.0. The *Administration Console Help* is greatly improved and is available from E-docs. Also, the following new guides have been added:

- *Using WebLogic Logging Services* at
  `http://e-docs.bea.com/wls/docs70/logging/index.html`

- *Upgrade Guide for WebLogic Server* at
  `http://e-docs.bea.com/wls/docs70/upgrade/index.html`

- *Creating and Configuring WebLogic Server Domains* at
  `http://e-docs.bea.com/wls/docs70/admin_domain/index.html`

- *WebLogic Security* at
  `http://e-docs.bea.com/wls/docs70/secintro/security.html`

- *Developer Tools,* at `http://e-docs.bea.com/wls/docs70/toolstable/tools.html`

- *Deployment documentation* at
  `http://e-docs.bea.com/wls/docs70/deployment.html`.

Examples

There have been many improvements made to the examples featured in WebLogic Server 7.0. Changes include the following:

- Several examples now run "out of the box." When you start your Examples Server a browser opens and allows you to try out some of the capabilities of WebLogic Server.

- Startup scripts to run the Examples server and Pet Store server have been improved to be easier to use.

- Many examples now include `ant` commands that simplify running them; in the `build.xml` file accompanying each example you will find the command line syntax that is scripted.

- The database system used by the WebLogic 7.0 samples has changed; the samples now use the PointBase® Server, a 100% Pure Java RDBMS.

Several new examples demonstrate the following features:

- Session replication with clustered servers

- BLOB/CLOB, Bands, Embedded Key, and Multi-table for EJB 2.0

- RMI-IIOP secure Tuxedo clients

- Javamail

- jCom

- JDBC Dynamic Data Source, JDBC Oracle extensions, and JDBC Dynamic Pool

- JMS using a message-driven beans with data sent in XML

- New Security demonstrating WebLogic's latest security implementation

- Servlets filters, lifecycle, and wrappers

- Using the WebLogic XML Streaming API

- Web Services with an EJB 2.0 stateless session bean, logging and more

Pet Store 1.3

The Pet Store sample application that is provided with WebLogic Server 7.0 has been upgraded to version 1.3. The Pet Store application is available from the Start menu and the source code is located in the *SAMPLES_HOME*\server\src\petstore directory, where *SAMPLES_HOME* is the location of all examples for the WebLogic Platform. By default, this location is c:\bea\weblogic700\samples.

WebLogic Server Tour

The WebLogic Server Tour is an overview of WebLogic Server that uses the Pet Store application to demonstrate features. The Tour is available from the Start menu.

Platform Support

See the *Platforms Support Page* at http://www.weblogic.com/platforms/index.html for the most accurate and current information regarding platform support.

Versions Compatibility for Standards and Libraries

The table below lists versions of standards and libraries included with or compatible with WebLogic Server 7.0.

Table 1-2 Standards and Libraries Compatible with WebLogic Server 7.0

Standard or Library	Version
org.w3c.dom interfaces	
org.xml.sax interfaces	
antlr	2.7.1
EJB	2.0 and 1.1
HTTP	1.1
JAAS	.0 Full
Java RMI	1.0
JavaMail	1.1
JAXP	1.1

Table 1-2 Standards and Libraries Compatible with WebLogic Server 7.0

JAX-RPC	1.0
JCA	1.0
JDBC	2.0.x
JMS	1.0.2
JMX	1.0
JMX	1.0
JNDI	1.2
JSP	1.2 and 1.1
JTA	1.0.1
LDAP	2
Oracle	• 8.1.7 • 9.0.1 • 9.2.0
OTS/JTA	1.2/1.0.1
RMI/IIOP	1.0
Servlet	2.3 and 2.2
SSL	3
X.509	3
Xerxes DOM and SAX	• DOM parser: Level 2.0 Core • SAX parser: Version 2.0
XML	org.apache.xalan Any version compatible with the version of Xerces XML parser.

Table 1-2 Standards and Libraries Compatible with WebLogic Server 7.0

XML parsers	org.apache.xerces
	Built-in version: 1.4.4
	Compatible with WebLogic Server 7.0.0.1 or previous:
	• 1.2.2
	• 1.2.3
	• 1.3.0
	• 1.3.1
	• 1.4.0
	• 2.0.0
	• 2.0.1
	Compatible with WebLogic Server 7.0 SP1 or later:
	• Any version of Xerces

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 7.0 includes the 1.3.1_02 JDK from Sun Microsystems.

WebLogic Javadoc

BEA uses the Javadoc tool to generate documentation for externally supported APIs. Javadoc is a tool from Sun used to document API of classes that have been declared within the public scope. A class declared within the public scope is visible to other classes outside of its own package. BEA uses a custom version of Javadoc that has been enhanced to distinguish between externally supported public classes and public classes that are intended for internal use only. However, Javadoc may generate indirect references to internal classes or methods in usage pages, inheritance trees, or in other supplemental lists. Such indirect references do not imply support for internal methods or classes. BEA supports only those APIs that are fully documented on the main Javadoc pages for the classes to which

they belong. In other words, when you select a package in the upper left-hand frame, all externally supported classes within that package will appear in the lower left-hand frame.

Creating Your Own MBeans

WebLogic Server provides hundreds of MBeans to configure and monitor WebLogic services such as the Security Service and JMS and deployable J2EE modules such as EJBs and Web applications. If you want to use additional MBeans to configure your applications or services, you can create your own MBeans.

If you want to create MBeans for a security provider, use the MBean-generation tools provided by the WebLogic Security Service. These tools are supported only for generating security-provider MBeans. For information on security providers, refer to *Developing Security Providers for WebLogic Server*.

If you want to create MBeans for any other managed resource on WebLogic Server, refer to the JMX specification (which you can download from http://java.sun.com/products/JavaManagement). MBeans that you create and register on a WebLogic Server instance can take advantage of the full set of JMX features as defined by the JMX specification. For more information about the WebLogic Server implementation of JMX, refer to *Programming WebLogic Management Services with JMX* at {http://e-docs.bea.com/wls/docs70/jmx/index.html.

Other Available Resources

Here are some pointers to useful information related to WebLogic Server 7.0. The hyperlinks require Internet access.

Fast Track Procedures

High-level procedures to help you quickly deploy an HTML file, JSPs, and servlets, are available at http://e-docs.bea.com/wls/docs70/quickstart/quick_start.html.

Pet Store 1.3

The Pet Store sample application demonstrates WebLogic Server features. It is available from the Start menu in the directory where you installed WebLogic Server 7.0.

Examples

Code examples, if installed, are located in the *SAMPLES_HOME*\server\src\examples directory of your WebLogic Server installation, where *SAMPLES_HOME* is the location of all examples for the

WebLogic Platform. By default, this location is `c:\bea\weblogic700\samples`. Examples are also available from the Start menu for Windows users.

Introduction

For an overview of WebLogic Server features and the J2EE application architecture, see *Introduction to WebLogic Server* at `http://e-docs.bea.com/wls/docs70/intro/index.html`.

Additional Documentation

The full documentation set for BEA WebLogic Server, including administration, programming, and reference guides, is provided on the BEA WebLogic Server Online Documentation CD-ROM and on the BEA Web site at `http://e-docs.bea.com/index.html`.

Newsgroups

BEA WebLogic Server newsgroups provide community support for BEA products. Information about BEA-related newsgroups can be found at `http://newsgroups.bea.com/` and `news://newsgroups.bea.com`.

Dev2Dev Online

The BEA site Dev2Dev Online provides resources to make your e-commerce development easier and faster. To reach Dev2Dev online, go to `http://developer.bea.com/`.

Known Issues

The following sections describe known issues, levels of support, notes, and general problems in WebLogic Server 7.0 and its service packs:

Known Issues in WebLogic Server 7.0 Service Pack 7

WebLogic Server 7.0 Service Pack 7 has the following known issues:

Core Server Known Issues

Change Request Number	Description	Release Fixed
CR174779	Requests from an existing session are directed to the primary server even after the primary server starts to shut down. **Workaround:** Refesh the page after a short duration.	

EJB Known Issues

Change Request Number	Description	Release Fixed
CR262695	Fully classified class names are not generated in the `ejb.xml` and `weblogic-cmp-rdbms-jar.xml` deployment descriptor files. Instead, only simple class names are generated for method param values. **Workaround:** When using unloadable classes in your EJBs, manually edit the deployment descriptor files to refer to those EJBs with their fully classified names.	

RMI Known Issues

Change Request Number	Description	Release Fixed
CR258722	WebLogic Server 6.1 has some limitations when acting as a client to WebLogic Server 7.0 SP7 over the IIOP protocol. This is because WebLogic Server 6.1 does not support class evolution, so only identical classes can be passed between WebLogic Server 6.1 and WebLogic Server 7.0 SP7. **Workaround:** Be aware of this issue.	
CR261149	Remote requests over IIOP that involve medium to large size Strings may be a little slower in WebLogic Server 7.0 SP7 when compared to 7.0SP6. This is due to changes in JDK 1.4.2_10, which is being used in WebLogic Server 7.0SP7. **Workaround:** Pass the `XX:+UseTLAB` JVM option flag. This flag turns on thread-local memory allocation. This flag is enabled by default only for the server VM on SPARC.	

Known Issues in WebLogic Server 7.0 Service Pack 6

WebLogic Server 7.0 Service Pack 6 has the following known issues:

Administration Console Known Issues

Change Request Number	Description	Release Fixed
CR213792	The help information for defining security roles and security policies for EJBs is missing from the Administration Console in this release of WebLogic Server. For information about assigning security roles and security policies to EJBs, see Securing WebLogic Resources.	
CR215117	The attribute, JavaVMVendor (of `weblogic.management.runtime.JVMRuntimeMBean`), was added to the Administration Console server monitoring versions page for WebLogic Server 7.0 Service Pack 6. There is currently no corresponding popup for it in the Administration Console online help. As a result, a 404 error occurs when an attempt is made to access the online help popup for this attribute.	

JDBC Known Issues

Change Request Number	Description	Release Fixed
CR192616	The API `getXAConnection()` through the net driver is not supported in the IBM DB2 UDB type 2 JDBC 2.0 driver. It is recommended that you use a different driver that supports the `getXAConnection()` API.	

Plug-ins Known Issues

Change Request Number	Description	Release Fixed
CR217777	Deadlock occurs in Apache HTTP Server plug-in under load.	
	Problem: When using the Apache HTTP Server plug-in for distributing client connection requests to WebLogic Server instances in a cluster, clients may be unable to connect to the cluster and the following message is logged in the Apache HTTP Server plug-in:	
	`[error] server reached MaxClients setting, consider raising the MaxClients setting`	
	This problem occurs under load.	
	Platform: All	
	Workaround: When configuring the Apache HTTP Server plug-in, set the value for the MaxClients parameter in the httpd.conf configuration file to be equivalent to the number of client connections that the cluster needs to be able to support.	
	For more information about configuring the Apache HTTP Server plug-in, see "Installing and Configuring the Apache HTTP Server Plug-In" in Using Web Server Plug-Ins with WebLogic Server at the following URL:	
	http://e-docs.bea.com/wls/docs81/plugins/apache.html	

Known Issues in WebLogic Server 7.0 Service Pack 5

WebLogic Server 7.0 Service Pack 5 has the following known issues:

Core Server Known Issues

Change Request Number	Description	Release Fixed
CR123714	Network classloading can fail when multiple servers, each with its own Network Interface Card, are operating in a single cluster or domain. Failures occur because in the JVMID of the server the `hostAddress` is always assumed to be the default channel address. The client uses `RJVMImpl.getCodeBase` to locate the URL from which to download the class, tries to connect to the default channel, and fails. In 8.1 the `hostAddress` of the JVMID is based on the channel the request comes from, and thus the `hostAddress` correctly registers the address of the channel.	8.1
CR079543, CR084568	On Unix, some server start scripts will fail if the name of the directory in which you install WebLogic Server (*BEAHOME*) contains a space.	

Cluster Known Issues

Change Request Number	Description	Release Fixed
CR168598	In WebLogic Server 7.0, the Managed Server starts in the same mode as the Administration Server even if you specify a different mode on Managed Server startup.	
	For example, if you start the Administration Server in development mode and then start the Managed Server with a command line option of -Dweblogic.ProductionModeEnabled=true, the Managed Server still starts in development mode.	
	The Managed Server starts up in production mode only when Administration Server has been started in production mode.	
CR135944, CR100373, CR130592, CR135864	Staged one-phase applications on a Managed Server in Managed Server Independence mode, that is, with the Administration Server down, do not deploy when the Managed Server is on a separate file system or machine from the Administration Server. When the Managed Server is brought up in MSI mode, a deployment error such as the following occurs:	8.1
	`<Feb 5, 2004 5:06:36 PM EST> <Error> <Management> <141006> <Application 474001:Name=DefaultWebApp,Type=Application not found at \home2\mks\bea70sp4\user_projects\474001\applicati ons\DefaultWebApp><Feb 5, 2004 5:06:36 PM EST> <Error> <Management> <141004> <IOException opening application 474001:Name=DefaultWebApp,Type=Application, loading from path \home2\mks\bea70sp4\user_projects\474001\applicati ons java.io.FileNotFoundException: File does not exist:`	
	`\home2\mks\bea70sp4\user_projects\474001\applicati ons\DefaultWebApp at weblogic.management.mbeans.custom.Application.admi nLoad(Application.java:627)`	
	This error occurs because there is no staging mode for one-phase applications, and the file is not cached (staged) on the Managed Server.	

Documentation Known Issues

Change Request Number	Description	Release Fixed
CR196210	Previous versions of this document and various other sample documents erroneously described using weblogic.management.Admin.getInstance().getAdminMBeanHome() as a way to look up the MBeanHome interface on the Administration Server. However, the weblogic.management.Admin class is not public. Instead of using this non-public class, use JNDI to retrieve MBeanHome. See Determining the Active Domain and Servers in Programming WebLogic Server JMX Services.	

EJB Known Issues

Change Request Number	Description	Release Fixed
CR112260	The `<security-permission>` element for an EJB was not displayed in the Administration Console, and the descriptor validation failed for the `security-permission` attributes. When the bean was deployed and its client called, it was possible to change the `java.vm.version` property.	

`weblogic-ejb-jar.xml` contained this stanza:

```
<security-permission>
    <description>Optional explanation goes
here</description>
    <security-permission-spec>
        grant { permission
java.util.PropertyPermission
"java.vm.vendor", "read"; } ;
    </security-permission-spec>
</security-permission> </weblogic-ejb-jar>
```

Workaround: A code change was made to allow `<security-permission>` to be viewed and modified in the Administration Console.

To correctly configure the desired security permission, it is necessary to perform the following steps:

1. Change the grant permission in `<security-permission-spec>` to `{ permission java.util.PropertyPermission "java.vm.vendor", "read"; }`

2. Add this line to the `%WL_HOME%\server\lib\weblogic.policy` file:
   ```
   add grant codeBase "file:/<Your
   user_projects dir>/YourDomain/lib/-"
   {permission java.security.AllPermission;
   } ;
   ```
   This is required because the dynamically generated stub's `classpath` is `lib`.

3. Start the server with the `-Djava.security.manager` option.

Deployment Known Issues

Change Request Number	Description	Release Fixed
CR099037	Configuring a default Web application for a cluster and then targeting or redeploying that Web application to either a single server within the cluster or to the cluster itself will result in a deployment failure.	

JDBC Known Issues

Change Request Number	Description	Release Fixed
CR173725	An invalid batch value exception occurred using Oracle 10g thin driver. Analysis revealed that the exception was caused by an Oracle thin driver limit for the batch size. The limit seemed to be 16383 (half the max value for a short). **Workaround:** Limit the batch size or use 920 driver. Oracle has stated that the bug has been fixed and is now being backported on top of the 10.1.0.2 drivers.	

JMS Known Issues

Change Request Number	Description	Release Fixed
CR108664	The queue browser returns an expired message even if the message had already expired at enumeration time.	8.1

Security Known Issues

Change Request Number	Description	Release Fixed
CR096349, CR133465	Please review the security advisory information at Security Advisory BEA04-43.01.	8.1
CR125639	The note in the `weblogic/security/SSL/SSLSocketFactory.java` file is incorrect. The note should read as follows:	

The note in the `weblogic/security/SSL/SSLSocketFactory.java` file is incorrect. The note should read as follows:

This SSLSocketFactory cannot be plugged into JSSE because the `getDefault()` method used by JSSE returns an instance initialized to use the old Phaos SSL implementation, instead of the Certicom SSL implementation currently used by the server. The Phaos implementation does not allow you to configure the trusted certificates, while Certicom's would be configured according to the server SSL trust configuration. If you really need to access the Certicom implementation from JSSE, you can extend the `javax.net.SocketFactory` class and implement its `getDefault()` method to return an instance of your class that wraps the SocketFactory instance returned by the `weblogic.security.SSL.SSLSocketFactory.getDefaultJSS()` method. The following code example shows how to extend the `javax.net.SocketFactory` class to access the Certicom implementation from JSSE:

```
public class MySSLSocketFactory extends
javax.net.SocketFactory {
private static MySSLSocketFactory defFactory;
private javax.net.SocketFactory factory; private
MySSLSocketFactory() {
factory = weblogic.security.SSL.SSLSocketFactory.
getDefaultJSSE();    }
public static javax.net.SocketFactory getDefault() {
if (defFactory == null) defFactory = new
MySSLSocketFactory();    return defFactory;  }
public Socket createSocket(InetAddress host, int port)
throws IOException {
return factory.createSocket(host, port);     }
// implement the other three abstract createSocket
// methods the same way.          ... }
```

WebLogic Tuxedo Connector Known Issues

Change Request Number	Description	Release Fixed
CR074356	When you configure a redirect for tBridge, the redirect configuration information appears at the wrong place in the `config.xml` file. The redirect configuration appears as a child of the tBridgeGlobal MBean. It should appear at the same level as the tBridgeGlobal MBean. **Workaround:** Manually edit the `config.xml` file.	8.1

Web Services Known Issues

Change Request Number	Description	Release Fixed
CR091230	On Microsoft Windows, the `clientgen` Ant task may overwrite entities whose names differ only in capitalization. This occurs because Windows is case insensitive.	

Known Issues in WebLogic Server 7.0 Service Pack 4

WebLogic Server 7.0 SP4 has the following known issues.

jDriver Known Issues

Change Request Number	Description	Release Fixed
CR094209	Using WebLogic Server applications with distributed transactions and Oracle Server 9.2.0.x causes problems with the Oracle thin driver 9.2.0.x. **Workaround**: Use the Oracle 9.2.0.3 driver with patch 2717235. Check the Oracle Web site and apply any necessary patches. In addition to the workaround, the following tuning technique may help alleviate the problem: • Set the number of connections in the JDBC configured connection pools to equal the number of default execute threads, so that WebLogic Server does not have to wait for a connection to complete an activity on an XA resource.	

JSP Known Issues

Change Request Number	Description	Release Fixed

CR061104	When you run the JSP compiler on Windows systems, output files names are always created with lower-case names. To prevent this behavior, and preserve the case used in classnames, set the system property `weblogic.jsp.windows.caseSensitive` to true. You can set the property at the command line when compiling a JSP using the following command:
	`java -Dweblogic.jsp.windows.caseSensitive=true`
	`weblogic.jspc *.jsp`
	Or include this command in your WebLogic Server startup scripts:
	`-Dweblogic.jsp.windows.caseSensitive=true`	

Web Services Known Issue

Change Request Number	Description	Release Fixed
CR092268	Using `clientgen` through a proxy from behind a firewall sometimes fails because the proxy user name and password are not sent.	

Known Issues in WebLogic Server 7.0 Service Pack 3

WebLogic Server 7.0 SP3 has the following known issues.

Browser Known Issues

Change Request Number	Description	Release Fixed
CR106615	When using Netscape version 4.79 on HP-UX 11.0 and 11i, a browser session may fail due to bus errors.	
CR107134	(JRockit) When using Oracle server version 8.1.7.3, running on Linux, and using JRockit as your JVM, creating a new EJB may result in an end-of-file error. The subsequent version of Oracle server, 8.1.7.4, does not present this problem.	
CR108206	An exception thrown by `libCsup.2` in HPCode is not handled by WebLogic Server jDriver code, possibly causing Weblogic Server to dump core. This problem occurs when there are three layers of code: HPCode, native code, and Java. The problem has been observed while running Oracle 9.2.0. **Workaround:** Use client libraries from Oracle 9.2.0.3.	
CR094639	A wildcard URL (/*) added to the security-constraint of an EJB changes the return from req.getSession(false) after a session timeout. When the security-constraint contains a single non-wildcard URL, the return from req.getSession(false) is a null. When the security-constraint contains a non-wildcard URL, AND an additional wildcard(/*) URL, the return from req.getSession(false) is a valid session even though timeout has occurred. Release fixed: WebLogic Server 7.0 SP5.	

EJB Known Issues

Change Request Number	Description	Release Fixed
CR082058	If an EJB is deployed as part of an Enterprise Application, you cannot redeploy the EJB without redeploying the entire application.	8.1
CR096713	Some `EJBCacheRuntime` object IDs do not return results when you run the `snmpwalk` command against them because their `RuntimeMBeans` have been removed but their object IDs still exist in the Management Information Base: 150.x.x EJB Entity Home Runtime, 458.x.x EJB Stateful Home Runtime, 170.x.x EJB Stateless Home Runtime, 137.x.x EJBCache Runtime, 136.x.x EJB Cache Monitor Runtime, 150.x.x EJB Entity Home Runtime, 160.x.x EJB Message Driven Runtime, 161.x.x EJB Pool Runtime, 458.x.x EJB Stateful Home Runtime, 170.x.x EJB Stateless Home Runtime, 455.x.x Connector Connection Pool Runtime, and 312.x.x JTA Statistics Runtime.	

Tools Known Issues

Change Request Number	Description	Release Fixed
CR092278	In order to redeploy a Web Application using the `weblogic.Deployer` utility, you must specify both the `-source` and `-target` options on the command line. If the `-source` and `-target` options are not specified, the tool indicates success but does not redeploy the Web Application.	

Web Services Known Issues

Change Request Number	Description	Release Fixed
CR101160	Complex data types in service-specific exceptions are not handled in strict accordance with the Java API for XML-Based RPC (JAX-RPC) 1.0 specification, as follows: • **Section 4.3.6 WSDL Fault**, *Service Specific Exception*: When the `clientgen` Ant task generates the JAX-RPC stubs from a WSDL that contains a `wsdl:fault` element whose message part is an `xsd:complexType`, the generated stubs incorrectly map the `xsd:complexType` to a simple Java type. This means that when a client application uses the stubs to invoke the Web Service, and the service throws the service-specific exception (rather than the a `javax.xml.rpc.SOAPFaultException`), the client application cannot access the full complex data from the exception. There is no workaround to this problem. • **Section 5.5.5 Methods**, *Exceptions*, fourth bullet: The WebLogic Web Service runtime incorrectly maps multiple fields in a service-specific Java exception class to a simple XML data type, rather than to an `xsd:complexType`. **Workaround**: Always throw a `javax.xml.rpc.SOAPFaultException`, rather than a service-specific exception, when implementing a WebLogic Web Service.	

Known Issues in WebLogic Server 7.0 Service Pack 2

WebLogic Server 7.0 SP2 contains the following known issues.

HPUX 11i Users Must Apply a Patch

An incorrect version of the `hpux11/libweblogicunix1.sl` library was bundled with the WebLogic Server 7.0 SP2 download. To obtain the correct library, HPUX 11i users must download a patch from the following location, and apply it:

http://commerce.bea.com/products/weblogicserver/suppljar.jsp

To apply the patch:

1. Download the patch.

2. De-archive it to `$WLHOME/weblogic700/server/lib`:

   ```
   jar -xvf CR102314_700sp2.jar
   ```

   This will replace `hpux11/libweblogicunix1.sl` with a new library.

3. Prepend the `.jar` file to your `CLASSPATH`.

Known Issue with JDK 1.4.1 Support and SSL

In order for JDK 1.4.1 clients to use SSL, the Java 2 SDK `jsse.jar` must be removed from the `jre\lib` directory:

```
Directory of C:\j2sdk1.4.1\jre\lib:

08/29/2002 01:10a 462,386 jsse.jar 1 File(s) 462,386 b
```

Administration Console Known Issues

Change Request Number	Description	Release Fixed
CR110791	When you add custom help pages to the Administration Console, use the Console Extension described in Extending the Administration Console at http://e-docs.bea.com/wls/docs70/console_ext/index.html.	N/A
	If you use the alternative method of placing custom help files in `console.war`, be sure to add `console.war` to WL_HOME/server/lib/weblogic.jar. Otherwise the custom files will be wiped out when console.war is regenerated at server startup.	
CR066112	Deployment procedures using the Administration Console have changed since the 6.x release. For more information, see WebLogic Server Application Deployment at `http://e-docs.bea.com/wls/docs70/programming/deploying.html`.	N/A
CR069828	You can no longer edit the Web Services deployment descriptor using the Administration Console.	N/A
CR069927	Selecting Server --> Monitoring --> Process Output and selecting any of the view options does not display the output. If you select the "View Node Manager Output" while Node Manager is running, Node Manager will crash.	7.0 SP3

Change Request Number	Description	Release Fixed
CR073164	You cannot use the Administration Console to edit a domain other than the current domain of the Administration Server.	
CR074339	Explanatory text for creating configuration objects is not displayed when using Netscape 4.7 to access the console.	8.1
CR094174	The Administration Console gives a `RuntimeError` with Internet Explorer version 5 when you select JDBC->MultiPool in the left panel of the Console.	

Core Server Known Issues

Change Request Number	Description	Release Fixed
CR048834	On Solaris, if you are using JDK 1.3.1, and if the timestamp for WebLogic log messages is in Greenwich Mean Time (GMT) regardless of your local time settings, you need to update `jre/lib/i18n.jar`. In some cases, this JAR file is missing the resources that the JRE uses to localize time stamps. **Workarounds:** • Fix the JRE install with a proper `i18n.jar`. • If the platform does not support the `i18n.jar`, then a custom ResourceBundle implementation can be placed in the CLASSPATH from where it will be loaded. • Ensure that the TimeZone is set properly for the install and it supports DST if applicable to the Locale.	
CR059543	If you use the Administration Console to update an XML Registry by setting the `DocumentBuilderFactory` text field to NULL, then subsequently view the same XML Registry, the `DocumentBuilderFactory` text field is empty. It should contain WebLogic Server's default `DocumentBuilderFactory`, `weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl`.	

Change Request Number	Description	Release Fixed
CR063799	Calling `getLogFileName` on `VirtualHostMBean` returns an incorrect path.	8.1
CR065299, CR067642	The Java HotSpot(TM) Server VM is not yet supported on WebLogic Server 7.0. Use the certified client version of this VM.	
CR066581	The WebLogic Server version of `javax.management.timer.Timer`, called `weblogic.management.timer.Timer`, was incompatible with the JMX timer. If you use the WebLogic class and instantiate it, you will have timers that use the WebLogic thread pools and honor the user identity when executing their listeners.	
CR066723	Managed Server discovery is off by default when you are in development mode. To turn it on, add `-Dweblogic.manager.discover=false` to your command line argument that starts WebLogic Server.	
CR067087	WebLogic Server currently does not provide a means to set security permissions within a `weblogic-applications.xml` file.	

CR067814	If you wish to use an LDAP browser to view the embedded LDAP server on WebLogic Server, try using the following input:
	`host: localhost`
	`port: 7003`
	`Anonymousbind`
	`LDAP Version 3`	

CR067868	When using the HotSpot JVM the server can hang when running large deployments. **Workaround:** Try setting your `-XX:MaxPermSize` to a higher figure.	

Change Request Number	Description	Release Fixed
CR068186	If you install WebLogic Server version 7.0 on a multi-homed, Windows 2000 system, configured servers will incorrectly resolve each IP address to the common DNS name for the system itself. For example, consider a multi-homed Windows 2000 machine named "MyWin2KMachine" and having IP addresses: 192.168.1.105 192.168.1.106 192.168.1.107 If you install multiple WebLogic Server instances on the system using IP addresses 192.168.105, 192.168.1.106, and 192.168.1.107 for the servers' Listen Address attributes, all servers will yield the common DNS name, "MyWin2KMachine," when resolving their IP addresses. This can lead to conflicts when accessing resources on remote servers. **Workarounds:** • Edit the c:/winnt/system32/drivers/hosts file with a text editor to explicitly name each IP address. For example: 192.168.1.105 MyWin2KMachine1 192.168.1.106 MyWin2KMachine2 192.168.1.107 MyWin2KMachine3 • If the machine's IP addresses have names registered with a DNS server, use those externally-available DNS names instead of IP addresses when configuring servers' Listen Address attributes. For example, configure servers using MyWin2KMachine1, MyWin2KMachine2, and MyWin2KMachine3 rather than using 192.168.105, 192.168.106, and 192.168.107. Note that this problem does not affect other software platforms, including Windows NT.	
CR068228	When you are not running in compatibility mode and you use the command line Administration tool, you may be prompted for a 'guest' password. The default password for "guest" is "guest".	
CR068247	The JSPs included with WebLogic Server 7.0 are not being pre-compiled before they are accessed for the first time. If you are deploying precompiled JSPs, WebLogic Server may recompile them to the /wlnotdelete/ directory.	

Change Request Number	Description	Release Fixed
CR068361	If you are migrating a 1.1 EJB from Bean Managed Persistence to Container Managed Persistence, selecting the bean and checking "Container-managed persistence" should generate Advanced and Automatic Key Generation options, but does not.	
CR069278	When using the command line to create a connection pool, in the properties string you can only escape the double quotes on platforms where the shell removes one set of quotes. **Workaround:** On platforms where the quotes are not removed during this process, you should use only one set of double quotes. You can also repeat the `property` argument for each connection property, which eliminates the need for quoting a string of properties. For example: `-property Property user=scott -property Password tiger -property URL jdbc:weblogic:oracle:ora817`	
CR069602	The JTA implementation in WebLogic Server was updated to comply with the JTA specification version 1.0.1a, including minor changes to the `Transaction.commit()` and `TransactionManager.setTransactionTimeout(int)` methods.	
CR072646	When using Netscape version 4.75 or older on Windows 2000, file associations may not be converted correctly from the same used for Internet Explorer. This results in the following behavior when opening About WebLogic Server 7.0, About Examples, and About Pet Store from the Windows Start menu: • An error message is displayed indicating failure to find target URI (even though the page opens in a browser). • If the browser is already running, the HTML page fails to display in the browser.	

Change Request Number	Description	Release Fixed
CR072940	If your `config.xml` file does not have the `<SecurityConfiguration>` element set, or is missing the credential and name attributes within that element, your server tries to create this attribute. If your `config.xml` file is writeable, your java client caches the value of this credential in its local JVM memory space. Then, for the lifetime of the JVM, the java client uses this credential. However, once the JVM gets re-started, it is denied access because it does not have the credential. **Workaround:** save your `config.xml` file once, when it has the credential already set.	
CR073395	There is a problem with Sun's Pet Store application when you try to run it with more than one client and use an Oracle database.	
CR073647	Currently only users in an Administration role can use the AdministrationPort.	8.1
CR073737	If you restart an Administration Server for an active domain, the Administration Server can fail to discover its Managed Servers. This occurs with the following configuration: A multihomed machine running a Windows operating system, in which the Administration Server and one or more processes use the same listen ports (but on different IP addresses). It is caused by a limitation of the Windows operating system. If a restarted Administration Server does not discover its Managed Server, commands that you issue to a domain via the Administration Server will fail. For example, a Managed Server that starts in Managed Server Independence (MSI) mode can fail to respond to shutdown commands after the Administration Server regains control.	
CR073793	You cannot deploy an application with untargeted modules if you have not set URIs for the untargeted modules in your `config.xml`.	8.1
CR074296	In some situations, the Administration Console allows users to create more than one WebLogic Tuxedo Connector `WTCResources` MBean. Only one `WTCResources` MBean is allowed for a given `WTCServer` MBean configuration. Users may need to remove the extra configuration and restart the WebLogic Tuxedo Connector.	8.1

Change Request Number	Description	Release Fixed
CR074452	If your JVM is throwing errors and WebLogic Server is under heavy load, try increasing your java memory heap. For example, modify the command that starts WebLogic Server by adding the following: `MEM_ARGS="-XX:MaxPermSize=128m -Xms512m -Xmx512m"`	
CR074969	You may encounter problems deploying EJBs without JNDI names. BEA recommends that users continue to deploy their EJBs with JNDI names as was done in earlier releases of WebLogic Server.	
CR085169	When using Iplanet 4.1SP9, the Java client can not read post data from a plug-in due to read_timeout from the java client using `socket.getInputStream().readLine()'` caused by a problem in `netbuf_getbytes()`. **Workaround:** Using Iplanet 6.0SP2 with WebLogic Server eliminates this problem.	
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), but may exist for other shells. The problem does not occur with the `bash` shell. **Workarounds:** • http://askbea-int.beasys.com/askbea/wls/S-15924.html provides a workaround (the `-Xrs` flag), but at the cost of losing thread dumps. • Use `"(startWeblogic >output 2>&1 &)"` instead of `"nohup startWeblogic &"`.	
CR110028, CR120181	Stuck thread detection in WebLogic Server 7.0 is limited to threads in the default execute queue. Stuck thread detection is done by `checkStuckThreads` method of `weblogic.t3.srvr.CoreHealthService`, and a private variable, `policyName`, which corresponds to the queue name for checking stuck threads, is statically set to "default."	

EJB Known Issues

Change Request Number	Description	Release Fixed
CR089847	Many 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. **Workaround:** To make WebLogic Server compliant with the EJB specification, 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`.	N/A
CR089630	When a table contains both BLOB and LONG RAW columns, OracleThinDriver requires that the LONG RAW column be read first from the stream even if BLOB occurs before LONG RAW in the select list. The EJB container currently does not handle this case and throws a `java.sql.SQLException` "Stream already closed". **Workaround:** Use JDriver.	
CR097323	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. 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 is deployed. As a result, the server instances in the cluster to which the EJB is 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 `Assertion` error is thrown in the RMI layer. **Workaround**: If you are deploying or redeploying an EJB to a single server instance in a cluster, compile the EJB before deploying it to ensure that the generated classes are copied to all server instances and available to all nodes in the cluster.	N/A

Examples and Pet Store Known Issues

Change Request Number	Description	Release Fixed

CR055626	The `login.jsp` file for the webapp security sample does not allow for URL encoding when a user logs in. The `login.jsp` file is in this location:
	
	`SAMPLES_HOME\server\src\examples\webapp\security`
	
	The `login.jsp` file has this code:
	
	`<form method="POST" action="j_security_check">`
	
	Here is an example of code that could be used to allow for URL encoding:
	
	`<FORM ACTION="<%=`
	`response.encodeURL("j_security_check") %>"`
	`METHOD=POST>`
		7.0 SP2

CR068009	Due to Microsoft VB Scripting Versioning, Excel 2000 or later is required to run the ZeroClient jCom example.	
CR071260	The password and username for the examples and Pet Store has been changed since WebLogic Server 7.0 Beta. Using `installadministrator/installadministrator` no longer works, use `weblogic` and `weblogic` instead.	N/A
CR073395	Pet Store has problems when multiple clients are using the same resources. Sun is working to address this problem.	

Change Request Number	Description	Release Fixed
CR074004	In the `package-summary.html` file in the `SAMPLES_HOME\server\src\examples\security` directory, the Security Examples Summary table makes references to the audit and cert samples. These samples have been removed from the WebLogic Server distribution because they used APIs that are being deprecated in WebLogic Server 7.0.	N/A
	Also, the descriptions of the SSLClient, JAAS, and Net samples should read as follows:	
	SSLClient: Demonstrates how to make outbound SSL connections and two-way SSL connections from a WebLogic Server that is acting as a client to another WebLogic Server or application server.	
	JAAS: Demonstrates use of the Java Authentication and Authorization Service by validating username and password, and then, if valid, executing the stateless session Trader EJB.	
	Net: Demonstrates writing connection filters that implement the `weblogic.security.net.ConnectionFilter` and `ConnectionFilterRulesListener` interfaces. Connection filters allow you to reject or accept client connections to the WebLogic Server environment based on the client's IP address and protocol and the local and remote port numbers.	
CR080288	There are problems with the RMI cluster example when using `ant run` to execute the client. The java call to the client includes incorrect quotation marks.	

Installation Known Issues

Change Request Number	Description	Release Fixed
CR085531	After successful upgrades and downgrades of a WebLogic Server via SmartUpdate, a subsequent upgrade results in the following error: `Installer is not compatible with the installed product. Your product directory contains installed components that are not available in the current installer. Please check your installer to ensure it matches the product distribution on your system.` **Workaround:** Contact your customer support representative for assistance.	

Interoperability Known Issues

Change Request Number	Description	Release Fixed
CR093343 and CR093347	When a Java client running on Java 2 SDK 1.4.1 accesses a server running on Java 2 SDK 1.3.1, the following errors occur: • The `getDate()` and `setDate()` methods return inconsistent results. • When calling `getTimeStamp()` or `setTimeStamp()` from the client, the client gets the "class not found" error. These are known issues with the Sun Java 2 SDK (Ticket #808980). Sun plans to fix these errors in Java 2 SDK 1.5. **Workaround**: Run the client application on SDK 1.3.1.	

Change Request Number	Description	Release Fixed
CR092507, CR095479	Remote methods throwing `MarshalException/UnmarshalException` when interoperating between 7.0 SP1 and 7.0 SP2 servers may result in unmarshalling problems. This occurs because `MarshalException` and `UnmarshalException` map to the correct completion statuses in SP2 but the incorrect completion statuses in SP1.	
CR095804	When a WebLogic Server 7.0 SP2 client attempts to get the Coordinator remote object from a WebLogic 8.1 server instance, a `ClassCastException` is thrown while casting the remote stub to the CoordinatorOneway interface. This prevents commit handoff to the 8.1 coordinating server.	7.0 SP3

jDriver Known Issues

Change Request Number	Description	Release Fixed
CR094209	Using WebLogic Server applications with distributed transactions with Oracle Server 9.2.0.x causes problems with the Oracle thin driver 9.2.0.x. **Workaround**: Use the Oracle 9.2.0.3 driver with patch 2717235. Check the Oracle Web site and apply any necessary patches. In addition to the workaround, the following tuning technique may help alleviate the problem: • Set the number of connections in the JDBC configured connection pools to equal the number of default execute threads, so that WebLogic Server does not have to wait for a connection to complete an activity on an XA resource.	

JMS Known Issues

Change Request Number	Description	Release Fixed
CR075438	A message-driven bean (MDB) that is deployed on a distributed topic, and which is targeted to a server instance that is hosting two members of the distributed topic, is also deployed on all the members of the distributed topic. This is the expected behavior because the MDB is pinned to the member destination name. Therefore, if a JMS server contains two members, there will be two MDBs deployed, one for each member.	

JRockit Known Issues

Change Request Number	Description	Release Fixed
CR093270	When running WebLogic Server 7.0 SP2 server on the JRockit JVM, in UDDI, publishing a new Web service to the private registry fails with this error message: "An error has occurred E_fatalError(10500): a serious technical error has occurred while processing the request." This is due to a bug in the UDDI implementation (AUDDI 1.0).	

JTA Known Issues

Change Request Number	Description	Release Fixed
CR089804	In a transaction, you cannot use connections from two Tx Data Sources that point to the same connection pool. When you attempt to get a connection from the second Tx Data Source, you will get an XA_PROTO error. Internally, WebLogic Server uses the connection pool name to identify the resource, not the data source name. When you get two separate connections from the same connection pool using different Tx Data Sources, they are treated as the same resource. In a transaction, start/enlist is called on the first Tx Data Source, then start/enlist is attempted on the second Tx Data Source. Because the server considers the resources to be the same, an XA_PROTO error is thrown because end/delist was not called before starting a new transaction. **Workarounds:** • Use a connection from only one Tx Data Source during the transaction. • Use two Tx Data Sources that point to different connection pools. There should be no performance degradation from either of these workarounds.	

Node Manager Known Issues

Change Request Number	Description	Release Fixed

CR074230	Applications deployed to WebLogic Server should perform file lookups relative to the Root Directory obtained with the `ServerMBean.getRootDirectory()` method (this defaults to the "." directory). For example, to perform a file lookup, use code similar to:
	
	```String rootDir = ServerMBean.getRootDirectory()\n //application root directory```
	
	```File f = new File(rootDir + File.separator +\n"foo.in");```
	
	rather than simply:
	
	```File f = new File("foo.in");```
	
	If an application is deployed to a server that is started using Node Manager, use the following method calls instead:
	
	```String rootDir //application root directory```
	
	```if ((rootDir = ServerMBean.getRootDirectory()) ==\n null) rootDir = ServerStartMBean.getRootDirectory();```
	```File f = new File(rootDir + File.separator + "foo.in");```
	
	The `ServerStartMBean.getRootDirectory()` method obtains the Root Directory value that you specified when configuring the server for startup using Node Manager. (This corresponds to the Root Directory attribute specified the Configuration->Remote Start page of the Administration Console.)
	
	A Managed Server started remotely by Node Manager does not recognize the `weblogic.RootDirectory` set on the Remote Start tab.
	
	To avoid this problem, enter an absolute path for Staging Directory on the Domain—>Servers—>Configuration—>Deployment tab. Do not enter a relative path. Enter the full path, including the WebLogic Server root directory.	

CR073078	When the Administration Server attempts to set up an SSL connection to Node Manager, it *does not* automatically search for certificate files in the Administration Server's root directory. If you store SSL certificate files in the Administration Server's root directory, make sure you configure the absolute path to the certificate files using the Connections->SSL tab in the Administration Console.	8.1

Change Request Number	Description	Release Fixed

CR075142	The Administration Server and Node Manager may simultaneously attempt to close a TCP socket, causing the following error message to appear in the Node Manager log file:
	
	`<Apr 18, 2002 5:32:55 PM EDT> <Error>`
	`<NodeManager@localhost:5555>`
	`<OutputHandler: done2 socket close failed, reason:`
	`java.net.SocketException:`
	`Connection reset by peer.>`
	
	This problem does not affect the operation of Node Manager or the Administration Server.	

CR094722	When Managed Servers in a cluster are booted, a `socketWrite` error is thrown on the Administration Server. **Workaround:** Ignore the problem or restart the managed node.	

Plug-in Known Issues

Change Request Number	Description	Release Fixed
CR095161	Under load, an Apache plug-in may receive a CONNECTION REFUSED error from a back-end WebLogic Server instance. This error was observed with Apache Server version 1.3.22 with the Apache plug-in and WebLogic Server running on Solaris 8.	

RMI Known Issues

Change Request Number	Description	Release Fixed
CR099693	A client accessing WebLogic Server through a firewall using the t3 protocol resulted in an error because the server expected to route a message over an uninitialized connection.	7.0 SP03
	The server's ExternalDNSName was specified. Upon initial connection, the client addressed the request to the server instance's ExternalDNSName. The JVMID in the response from the server instance required the server's internal IP address and internal DNS name. Subsequent requests from the client were not recognized.	
	The problem occurs if clients are connecting over t3 through a firewall. The workaround is to add an entry to the external DNS servers that maps the server instance's internal DNS name to its external IP address.	

Security Known Issues

For information about BEA security advisories, refer to the BEA Advisories & Notifications page on the dev2dev Web site. On this page, you can download security-related patches and register to receive notifications of newly available security advisories. You can access the Advisories & Notifications page with the following URL: http://dev2dev.bea.com/advisories.

BEA has established an e-mail address (security-report@bea.com) to which you can send reports of any possible security issues in BEA products.

The following table summarizes security-related known issues for this service pack release:

Change Request Number	Description	Release Fixed
CR086158	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-22.jsp.	
CR092546	The proper syntax for a role name must be an XML_NMTOKEN as defined in the specification at: http://www.w3.org/TR/REC-xml#NT-Nmtoken	
CR093652	In order for clients built with JDK 1.4.1 to use SSL, the Java 2 SDK jsse.jar file must be removed from the *JAVA_HOME*\jre\lib directory.	
CR094566	To use the nCipher JCE provider with this release of WebLogic Server, clients must be built using JDK 1.3.1.	
CR096589	weblogic.servlet.security.ServletAuthentication. strong() is broken and fails with a class cast exception. **Workaround**: Install the following patch: CR096589_70sp2.jar.	7.0 SP3

Third-Party Tools Known Issues

Change Request Number	Description	Release Fixed
CR071527	There is a problem persisting sessions with Resonate—a third-party load balancing tool that uses WebLogic Server as a backend server. Because Resonate does not handle case insensitivity, if URLEncoding sends a cookie as jsessionid instead of JSESSIONID, Resonate does not detect the cookie.	

Tools Known Issues

Change Request Number	Description	Release Fixed
CR064893	The following deployment descriptor elements cannot be edited in WebLogic Builder: `ejb-jar.xml` elements: • `cascade-delete (EB/Advanced)` • `weblogic ejb xml` • `transaction-isolation` • `description, display-name, large-icon, small-icon` • `exclude-list` • `unchecked` • `use-caller-identity` `weblogic-cmp.xml elements:` • `enable-dynamic-queries (EB/Advanced)` • `isolation-level (Methods/Transactions)` • `multiple table mapping` • `db-cascade-delete` • `field-group, group-name (*)` • `relationship-caching, caching-name, caching-element` • `validate-db-schema-with` • `database-type`	
CR065250	WebLogic Builder may fail to access a database on a server if the driver classes for the database are not on the client.	
CR066466	WebLogic Builder does not provide the option to cancel operations.	
CR066668	On Linux, WebLogic Builder's file browser opens at root instead of current directory.	
CR068152	The operations Add, Edit, and Delete can be slow in the EJB elements of WebLogic Builder.	8.1
CR068165	In the Add dialog in Resource Links, in the WebLogic Settings, J2EE Links, WebLogic Builder incorrectly offers a selection of application scoped data resources.	
CR068620	Table columns may not resize properly in WebLogic Builder.	8.1

Change Request Number	Description	Release Fixed
CR069568	WebLogic Builder cannot map multiple roles to a method in EJB method permission.	
CR071833, CR072541	WebLogic Builder support for RARs does not include generation of deployment descriptors.	
CR072700, CR074591	If you use WebLogic Builder's deployment dialog box to undeploy an application, the dialog will not show the undeployment. This dialog may also report that applications whose deployment has failed are deployed.	7.0 SP3
CR073975	WebLogic Builder incorrectly allows you to add a transaction without specifying a method or a transaction attribute.	7.0 SP3
CR074210	Optimistic concurrency strategy does not work in WebLogic Builder.	8.1
CR074322, CR074856	In WebLogic Builder, canceling an edit action may not work.	8.1
CR074566	Finders or container-managed persistence values modified through WebLogic Builder's BeanGrid/Edit (like include-update) may not be represented in the individual panels.	8.1
CR074622	In editing a container-managed persistence field using the WebLogic Builder's pop-up dialog, if you set column type to OracleCLOB and then click on the CMP tree node, the column type change does not show up on the CMP panel.	
CR074728	WebLogic Builder does not allow changing a method to a local method.	
CR074754	Editing some transaction attributes may cause WebLogic Builder to create duplicate descriptors.	
CR074961	If you use WebLogic Builder to generate `web.xml` descriptors, it creates a security-role "guest" which pertains to the pre-7.0 security protocol.	
CR075129	If you switch a stateless session bean to stateful, WebLogic Builder may not show that the module is dirty.	
CR075228	WebLogic Builder may throw an assertion error if you delete a transaction or permission method.	8.1

Change Request Number	Description	Release Fixed
CR075586	If you want to use WebLogic Builder with the WebLogic Server Examples Server or examples, you should run setExamplesEnv (located at `%SAMPLES_HOME%\server\config\examples`) before starting WebLogic Builder, so that files needed to run the PointBase database server will be added to the classpath. If you do not, you will get database error messages.	
CR075753	Some descriptor elements may pass through WebLogic Builder validation and then be found to be in error by `ejbc`.	
CR075924	In some cases, WebLogic Builder may not properly generate descriptor files for nested WAR files in EARs.	8.1
CR075936	WebLogic Builder may generate an illegal value for the element `estimated-bean-size`.	8.1

WebLogic Tuxedo Connector Known Issues

Change Request Number	Description	Release Fixed
CR067275	Load balancing and failover are not supported with WebLogic Tuxedo Connector in a cluster using RMI over IIOP.	
CR079630	Most users 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 to 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 "1234567890ABCDEFGHIJKLMNOPQRSTUV":

```
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.

Web Services Known Issues

Change Request Number	Description	Release Fixed
CR067190	WebLogic Web services do not support XML Schemas that contain complexTypes and elements with the same name.	

CR072890	The command line version of the `servicegen` Ant task does not support the following options that correspond to the Ant task attributes:
	
	`-overwrite`
	`-contextURI`
	`-includeEJBs`
	`-excludeEJBs`
	`-generateTypes`
	
	The command line version of the `clientgen` Ant task doe not support the following options that correspond to the Ant task attributes:
	
	`-autotype`
	`-overwrite`
		8.1

CR073661	WebLogic Web services that are described with an invalid `web-services.xml` file incorrectly deploy on WebLogic Server, although they are not accessible by client applications trying to invoke them.	
CR083493	The messages and multicomponent examples in the Web Services examples package do not contain the code needed to generate a downloadable client.jar file.This omission does not affect the functionality of the example which work with the provided client.	
CR094688	Using the 7.0 WebServices Portable Stubs client `.jar` may result in a `java.lang.NoClassDefFoundError`. **Workaround**: Run the client using the full CLASSPATH and provide the difference between the classes to the VersionMaker utility.	

Change Request Number	Description	Release Fixed
CR095629	When using the new `weblogic.webservice.cachewsdl` flag to optimize memory use, a bogus warning message displays in the server log: `<Warning> <Management> <141087> <Unrecognized property: webservice.cachewsdl>` This warning is bogus; in spite of this message, the flag takes effect.	
CR101160	Complex data types in service-specific exceptions are not handled in strict accordance with the Java API for XML-Based RPC (JAX-RPC) 1.0 specification, as follows: • **Section 4.3.6 WSDL Fault**, *Service Specific Exception*: When the `clientgen` Ant task generates the JAX-RPC stubs from a WSDL that contains a `wsdl:fault` element whose message part is an `xsd:complexType`, the generated stubs incorrectly map the `xsd:complexType` to a simple Java type. This means that when a client application uses the stubs to invoke the Web Service, and the service throws the service-specific exception (rather than the a `javax.xml.rpc.SOAPFaultException`), the client application cannot access the full complex data from the exception. There is no workaround to this problem. • **Section 5.5.5 Methods**, *Exceptions*, fourth bullet: The WebLogic Web Service runtime incorrectly maps multiple fields in a service-specific Java exception class to a simple XML data type, rather than to an `xsd:complexType`. The workaround is to always throw a `javax.xml.rpc.SOAPFaultException`, rather than a service-specific exception, when implementing a WebLogic Web Service.	

XML Known Issues

Change Request Number	Description	Release Fixed
CR070547	When the XML Streaming API is used on Solaris, some character references do not resolve to the same character as on NT.	

Known Issues in WebLogic Server 7.0 Service Pack 1

This section covers known issues in WebLogic Server 7.0 Service Pack 1.

Administration Console Known Issues

Change Request Number	Description	Release Fixed
CR066112	Deployment procedures using the Administration Console have changed since the 6.x release. For more information, see WebLogic Server Application Deployment at `http://e-docs.bea.com/wls/docs70/programming/deploying.html`.	N/A
CR069828	You can no longer edit the Web Services deployment descriptor using the Administration Console.	N/A
CR069927	Selecting Server --> Monitoring --> Process Output and selecting any of the view options does not display the output. If you select the "View Node Manager Output" while Node Manager is running, Node Manager will crash.	
CR073164	You cannot use the Administration Console to edit a domain other than the current domain of the Administration Server.	
CR074339	Explanatory text for creating configuration objects is not displayed when you use Netscape 4.7 to access the console.	8.1
CR074677	When you upgrade a domain created under version 6.x of WebLogic Server, the user "system" no longer explicitly grants access to the Administration Console. To access the console, a user must be a member of the Administrators Group.	7.0 SP2

Core Server Known Issues

Change Request Number	Description	Release Fixed
CR110058	The `MalformedURLException` can result when the dynamic group attribute `memberURL` contains a string value with a reserved character such as a forward slash '/') in the filter portion of the LDAP URL. Such a URL can result when the filter portion resolves to an object in the LDAP Directory Server that contains the forward slash character. **Workaround:** Avoid the use of reserved characters ':','?','/') in object names in your LDAP directory if your directory makes use of dynamic groups. The `netscape.ldap.LDAPUrl` utility class open source) cannot handle these characters.	
CR030981	The JDBC t3 Driver, which was deprecated in WebLogic Server 6.0, was removed from this release.	N/A
CR048834	On Solaris, if you are using JDK 1.3.1, and if the timestamp for WebLogic log messages is in GMT regardless of your local time settings, you need to update `jre/lib/i18n.jar`. In some cases, this JAR file is missing the resources that the JRE uses to localize time stamps. **Workaround:** To fix this there are several alternatives. • Fix the JRE install with a proper i18n.jar. • If the platform does not support i18n.jar then a custom ResourceBundle implementation can be placed in the CLASSPATH from where it will be loaded. • Ensure that the TimeZone is set properly for the install and it supports DST if applicable to the Locale.	
CR059543	If you use the Administration Console to update an XML Registry by setting the `DocumentBuilderFactory` text field to NULL, then subsequently view the same XML Registry, the `DocumentBuilderFactory` text field is empty. It should contain WebLogic Server's default `DocumentBuilderFactory`, `weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl`.	
CR063799	Calling `getLogFileName` on `VirtualHostMBean` returns an incorrect path.	8.1
CR065299, CR067642	The Java HotSpot(TM) Server VM is not yet supported on WebLogic Server 7.0. Use the certified client version of this VM.	

Change Request Number	Description	Release Fixed
CR066581	WebLogic Server has its own version of `javax.management.timer.Timer`, called `weblogic.management.timer.Timer`. If you use the WebLogic class and instantiate it, you will have timers that use the WebLogic thread pools and honor the user identity when executing their listeners.	
CR066723	Managed Server discovery is off by default when you are in development mode. To turn it on, add `-Dweblogic.manager.discover=false` to your command line argument that starts WebLogic Server.	
CR067087	WebLogic Server currently does not provide a means to set security permissions within a `weblogic-applications.xml` file.	
| CR067814 | If you wish to use an LDAP browser to view the embedded LDAP server on WebLogic Server, try using the following input:<br>```
host: localhost
port: 7003
Anonymousbind
LDAP Version 3
``` | |
| CR067868 | When using the HotSpot JVM the server can hang when running large deployments.
Workaround: Try setting your `-XX:MaxPermSize` to a higher figure. | |

Change Request Number	Description	Release Fixed

CR068186	If you install WebLogic Server version 7.0 on a multi-homed, Windows 2000 system, configured servers will incorrectly resolve each IP address to the common DNS name for the system itself. For example, consider a multi-homed Windows 2000 machine named "MyWin2KMachine" and having IP addresses:
	
	`192.168.1.105`
	`192.168.1.106`
	`192.168.1.107`
	
	If you install multiple WebLogic Server instances on the system using IP addresses 192.168.105, 192.168.1.106, and 192.168.1.107 for the servers' Listen Address attributes, all servers will yield the common DNS name, "MyWin2KMachine," when resolving their IP addresses. This can lead to conflicts when accessing resources on remote servers.
	
	Workaround: Do one of the following:
	
	• Edit the `c:/winnt/system32/drivers/hosts` file with a text editor to explicitly name each IP address. For example:
	
	`192.168.1.105 MyWin2KMachine1`
	`192.168.1.106 MyWin2KMachine2`
	`192.168.1.107 MyWin2KMachine3`
	
	• If the machine's IP addresses have names registered with a DNS server, use those externally-available DNS names instead of IP addresses when configuring servers' Listen Address attributes. For example, configure servers using MyWin2KMachine1, MyWin2KMachine2, and MyWin2KMachine3 rather than using 192.168.105, 192.168.106, and 192.168.107. Note that this problem does not affect other software platforms, including Windows NT.	

CR068228	When you are not running in compatibility mode and you use the command line Administration tool, you may be prompted for a 'guest' password. The default password for "guest" is "guest".	
CR068247	The JSPs included with WebLogic Server 7.0 are not being pre-compiled before they are accessed for the first time. If you are deploying precompiled JSPs, WebLogic Server may recompile them to the `/wlnotdelete/` directory.	
CR068361	If you are migrating a 1.1 EJB from Bean Managed Persistence to Container Managed Persistence, selecting the bean and checking "Container-managed persistence" should generate Advanced and Automatic Key Generation options, but does not.	

Change Request Number	Description	Release Fixed
CR069278	When using the command line to create a connection pool, in the properties string you can only escape the double quotes on platforms where the shell removes one set of quotes. On platforms where the quotes are not removed during this process, you should use only one set of double quotes. You can also repeat the `property` argument for each connection property, which eliminates the need for quoting a string of properties. For example: ```-property Property user=scott -property Password tiger -property URL jdbc:weblogic:oracle:ora817```	
CR069602	The JTA implementation in WebLogic Server was updated to comply with the JTA specification version 1.0.1a, including minor changes to the `Transaction.commit()` and `TransactionManager.setTransactionTimeout(int)` methods.	
CR070429	There is currently no way to re-boot with the last known good COMMO configuration.	7.0 SP2
CR071954	In 6.x versions, the `weblogic.management.Admin` class had the static method, `getActiveDomain()`. In Weblogic Server 7.0, `getActiveDomain()` is no longer static. In its place, use `Admin.getInstance().getAdminMBeanHome().getActiveDomain();` For an example of how to use this alternative API, see *Determining the Active Domain and Servers* in *Using WebLogic Server JMX Services*.	
CR072646	When you use Netscape version 4.75 or earlier on Windows 2000, file associations may not be converted correctly from the same used for Internet Explorer. This results in the following behavior when opening About WebLogic Server 7.0, About Examples, and About Pet Store from the Windows Start menu: • An error message is displayed indicating failure to find target URI (even though the page opens in a browser). • If the browser is already running, the HTML page fails to display in the browser.	

Change Request Number	Description	Release Fixed
CR072940	If your `config.xml` file does not have the `<SecurityConfiguration>` element set, or is missing the credential and name attributes within that element, your server tries to create this attribute. If your `config.xml` file is writeable, your java client caches the value of this credential in its local JVM memory space. Then, for the lifetime of the JVM, the java client uses this credential. However, once the JVM gets re-started, it is denied access because it does not have the credential. **Workaround:** save your `config.xml` file once, when it has the credential already set.	
CR073395	There is a problem with Sun's Pet Store application when you try to run it with more than one client and use an Oracle database.	
CR073647	Currently only users in an Administration role can use the AdministrationPort.	8.1
CR073737	If you restart an Administration Server for an active domain, the Administration Server can fail to discover its Managed Servers. This occurs with the following configuration: A multihomed machine running a Windows operating system, in which the Administration Server and one or more processes use the same listen ports (but on different IP addresses). It is caused by a limitation of the Windows operating system. If a restarted Administration Server does not discover its Managed Server, commands that you issue to a domain via the Administration Server will fail. For example, a Managed Server that starts in Managed Server Independence (MSI) mode can fail to respond to shutdown commands after the Administration Server regains control.	
CR073793	You cannot deploy an application with untargeted modules if you have not set URIs for the untargeted modules in your `config.xml`.	8.1
CR074296	In some situations, the Administration Console allows users to create more than one WebLogic Tuxedo Connector `WTCResources` MBean. Only one `WTCResources` MBean is allowed for a given `WTCServer` MBean configuration. Users may need to remove the extra configuration and restart the WebLogic Tuxedo Connector.	8.1
CR074452	If your JVM is throwing errors and WebLogic Server is under heavy load, try increasing your java memory heap.For example, modify the command that starts WebLogic by adding the following: `MEM_ARGS="-XX:MaxPermSize=128m -Xms512m -Xmx512m"`	

Change Request Number	Description	Release Fixed
CR074969	You may encounter problems deploying EJBs without JNDI names. BEA recommends that users continue to deploy their EJBs with JNDI names as was done in earlier releases of WebLogic Server.	
CR085169	When using Iplanet 4.1SP9, the Java client can not read post data from a plug-in due to read_timeout from the java client using `socket.getInputStream().readLine()'` caused by a problem in `netbuf_getbytes()`. **Workaround:** Use Iplanet 6.0SP2 with WebLogic Server.	
CR086784	You can encounter problems with deadlocks when starting WebLogic Server 7.0 SP1 on Windows 2000 with Sun Microsystem's JVM 1.3.1_03. These problems occur when using the JVM's over-optimizations. **Workaround:** use the JVM's `-XX:-UseCompilerSafepoints` option to turn off the over-optimizations.	

EJB Known Issues

Change Request Number	Description	Release Fixed

CR124381	An EAR file generates an put-of-memory error during deployment. The exception begins:
	
	`<Sep 30, 2003 8:28:41 AM MDT> <Info> <Deployer> <BEA-149060> <Module ttri_depl.jar of application OSS_TTRI successfully transitioned from prepared to unprepared on server myserver.>`
	
	`<Sep 30, 2003 8:28:41 AM MDT> <Info> <Deployer> <BEA-149033> <failed application OSS_TTRI on myserver>`
	
	`<Sep 30, 2003 8:28:41 AM MDT> <Info> <Management> <BEA-140009> <Configuration changes for the domain have been saved to the repository.>`
	
	Workaround: reduce the quantity setting for `max-beans-in-free-pool` in `weblogic-ejb-jar.xml`, for instance from:
	
	`<max-beans-in-free-pool>2147483647</max-beans-in-free-pool>`
	
	to
	
	`<max-beans-in-free-pool>100</max-beans-in-free-pool>`	

CR089847	Many 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. 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`.	N/A

Change Request Number	Description	Release Fixed
CR089630	In the case where a table contains both BLOB and LONG RAW columns, OracleThinDriver requires that the LONG RAW column be read first from the stream even if BLOB occurs before LONG RAW in the select list.	
	The EJB container currently does not handle this case and throws a `java.sql.SQLException` "Stream already closed".	
	Workaround: Use JDriver.	
CR097323	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.	
	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 `Assertion` error is thrown in the RMI layer.	
	Workaround: If you are deploying or redeploying an EJB to a single server instance in a cluster, compile the EJB with `ejbc` before deploying it, to ensure that the generated classes copied to all server instances available to all nodes in the cluster.	

Examples and Pet Store Known Issues

Change Request Number	Description	Release Fixed

CR055626	The `login.jsp` file for the `webapp security` sample does not allow for URL encoding when a user logs in. The `login.jsp` file is in this location:
	
	`SAMPLES_HOME\server\src\examples\webapp\security`
	
	The `login.jsp` file has this code:
	
	`<form method="POST" action="j_security_check">`
	
	Here is an example of code that could be used to allow for URL encoding:
	
	`<FORM ACTION="<%= response.encodeURL("j_security_check") %>" METHOD=POST>`
		7.0 SP2

CR068009	Due to Microsoft VB Scripting Versioning, Excel 2000 or later is required to run the ZeroClient jCom example.	
CR071260	The password and username for the examples and Pet Store has been changed since WebLogic Server 7.0 Beta. Using `installadministrator` and `installadministrator` no longer works, use `weblogic` and `weblogic` instead.	N/A
CR073395	Pet Store has problems when multiple clients are using the same resources. Sun is working to address this problem.	

Change Request Number	Description	Release Fixed
CR074004	In the `package-summary.html` file in the `SAMPLES_HOME\server\src\examples\security` directory, the Security Examples Summary table makes references to the audit and cert samples. These samples have been removed from the WebLogic Server distribution because they used APIs that are being deprecated in WebLogic Server 7.0. Also, the descriptions of the SSLClient, JAAS, and Net samples should read as follows: **SSLClient:** Demonstrates how to make outbound SSL connections and two-way SSL connections from a WebLogic Server that is acting as a client to another WebLogic Server or application server. JAAS: Demonstrates use of the Java Authentication and Authorization Service by validating username and password, and then, if valid, executing the stateless session Trader EJB. **Net:** Demonstrates writing connection filters that implement the `weblogic.security.net.ConnectionFilter` and `ConnectionFilterRulesListener` interfaces. Connection filters allow you to reject or accept client connections to the WebLogic Server environment based on the client's IP address and protocol and the local and remote port numbers.	N/A

Change Request Number	Description	Release Fixed
CR080288	There are problems with the RMI cluster example when using `ant run` to execute the client. The java call to the client includes incorrect quotation marks.	
CR085521	The package-summary.html file of the r3client Web service example is incomplete. This example is located in the SAMPLES_HOME\server\src\examples\webservices\r3client directory. The example illustrates how to run the WebLogic Web service client to exercise the SOAPBuilders Interoperability Lab Round 3 suite of tests. To build the example, follow these steps: 1. Set up your development shell as described in Quick Start. 2. Change to the SAMPLES_HOME\server\src\examples\webservices\r3client directory where SAMPLES_HOME refers to the examples WebLogic Server domain directory. 3. Assemble and compile the example by executing the Java ant utility at the command line: prompt> ant 4. Compile the results into an HTML table by executing the following Ant task at the command line: prompt> ant compileResults **Note:** You do not need to start WebLogic Server because the application invokes Web services from the SOAPBuilders Web site rather than one running on WebLogic Server. To check the output, open the following file in your browser to view the results of the tests in an HTML table: SAMPLES_HOME\server\src\examples\webservices\r3client\results\summary.html	7.0 SP2

Installation Known Issues

Change Request Number	Description	Release Fixed
CR085531	After successfully upgrading and downgrading WebLogic Server via SmartUpdate, a subsequent upgrade results in the following error: `Installer is not compatible with the installed product. Your product directory contains installed components that are not available in the current installer. Please check your installer to ensure it matches the product distribution on your system.` **Workaround:** Contact your customer support representative for assistance.	

JMS Known Issues

Change Request Number	Description	Release Fixed
CR066079	WebLogic JMS is now fully compliant with Sun Microsystems' JMS Specification version 1.0.b. Therefore, the message property name should not contain spaces.	
CR075438	A message-driven bean (MDB) that is deployed on a distributed topic, and which is targeted to a server instance that is hosting two members of the distributed topic, is also deployed on all the members of the distributed topic. This is the expected behavior because the MDB is pinned to the member destination name. Therefore, if a JMS server contains two members, there will be two MDBs deployed, one for each member.	

Node Manager Known Issues

Change Request Number	Description	Release Fixed

CR071591	Node Manager detects Managed Server failures by trying to initiate an HTTP connection. If Node Manager receives a connection failure, the connection exception can be observed in Node Manager output:
	
	`<Mar 19, 2002 2:30:04 PM EST> <Warning> <net> <000900>`
	`<Could not open connection`
	`java.net.ConnectException: Connection refused:`
	`connect`
		7.0 SP2

CR073078	When the Administration Server attempts to set up an SSL connection to Node Manager, it *does not* automatically search for certificate files in the Administration Server's root directory. **Workaround:** If you store SSL certificate files in the Administration Server's root directory, make sure you configure the absolute path to the certificate files using the Connections->SSL tab in the Administration Console.	8.1

Change Request Number	Description	Release Fixed

CR099693	In 7.0 (G.A., SP01, and SP02), a client accessing WebLogic Server through a firewall using the t3 protocol, resulted in this error on the server:
	
	`####<Feb 17, 2003 10:15:59 AM EST> <Error> <RJVM> <ewibeasbx01.merck.com> <edmServer> <ExecuteThread: '14' for queue: 'default'> <kernel identity> <> <000506> <Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection:`
	
	`'JVMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,7002,7001,7002,-1]:edm:edmServer' to: '0S:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0',flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''>`
	
	The server's `ExternalDNSName` was specified. Upon initial connection, the client addressed the request to the server instance's `ExternalDNSName`. The JVMID in the response from the server instance required the server's internal IP address and internal DNS name. Subsequent requests from the client were not recognized.
	
	This is a known problem in WebLogic Server 7.0 that is resolved in SP03. The problem occurs if clients are connecting over t3 through a firewall.
	
	Workaround: Add an entry to the external DNS servers that maps the server instance's internal DNS name to its external IP address.
		7.0 SP03

Change Request Number	Description	Release Fixed

CR074230	Applications deployed to WebLogic Server should perform file lookups relative to the Root Directory obtained with the `ServerMBean.getRootDirectory()` method (this defaults to the "." directory). For example, to perform a file lookup, use code similar to:
	
	``
	String rootDir = ServerMBean.getRootDirectory();
	 //application root directory
	File f = new File(rootDir + File.separator +
	 "foo.in");
	``
	
	rather than simply:
	
	``
	File f = new File("foo.in");
	``
	
	If an application is deployed to a server that is started using Node Manager, use the following method calls instead:
	
	``
	String rootDir //application root directory
	if ((rootDir = ServerMBean.getRootDirectory()) ==
	 null) rootDir =
	 ServerStartMBean.getRootDirectory();
	File f = new File(rootDir + File.separator +
	 "foo.in");
	``
	
	The `ServerStartMBean.getRootDirectory()` method obtains the Root Directory value that you specified when configuring the server for startup using Node Manager. (This corresponds to the Root Directory attribute specified the Configuration->Remote Start page of the Administration Console.)
	
	A Managed Server started remotely by Node Manager does not recognize the `weblogic.RootDirectory` set on the Remote Start tab.
	
	To avoid this problem, enter an absolute path for Staging Directory on the Domain—>Servers—>Configuration—>Deployment tab. Do not enter a relative path. Enter the full path, including the WebLogic Server root directory.	

Change Request Number	Description	Release Fixed

CR075142	The Administration Server and Node Manager may simultaneously attempt to close a TCP socket, causing the following error message to appear in the Node Manager log file:
	
	`<Apr 18, 2002 5:32:55 PM EDT> <Error>`
	`<NodeManager@localhost:5555>`
	`<OutputHandler: done2 socket close failed, reason:`
	`java.net.SocketException:`
	`Connection reset by peer.>`
	
	This problem does not affect the operation of Node Manager or the Administration Server.	

RMI Known Issues

Change Request Number	Description	Release Fixed

CR099693	In 7.0 (G.A., SP01, and SP02), a client accessing WebLogic Server through a firewall using the t3 protocol, resulted in this error on the server:
	
	`####<Feb 17, 2003 10:15:59 AM EST> <Error> <RJVM> <ewibeasbx01.merck.com> <edmServer> <ExecuteThread: '14' for queue: 'default'> <kernel identity> <> <000506> <Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection:`
	
	`'JVMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,700 2,7001,7002,-1]:edm:edmServer' to: '0S:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0',flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''>`
	
	The server's `ExternalDNSName` was specified. Upon initial connection, the client addressed the request to the server instance's `ExternalDNSName`. The JVMID in the response from the server instance the server's internal IP address and internal DNS name. Subsequent requests from the client were not recognized.
	
	This is a known problem in WebLogic Server 7.0 that is resolved in SP03. The problem occurs if clients are connecting over t3 through a firewall. The workaround is to add an entry to the external DNS servers that maps the server instance's internal DNS name to its external IP address.
		7.0 SP03

Security Known Issues

For information about BEA security advisories, refer to the BEA Advisories & Notifications page on the dev2dev Web site. On this page, you can download security-related patches and register to receive notifications of newly available security advisories. You can access the Advisories & Notifications page with the following URL: http://dev2dev.bea.com/advisories.

BEA has established an e-mail address (security-report@bea.com) to which you can send reports of any possible security issues in BEA products.

The following table summarizes security-related known issues for this service pack release:

Change Request Number	Description	Release Fixed
CR135489	When configuring custom security providers, you put the MBean JAR file for the provider in the **WL_HOME\lib\mbeantypes** directory. Placing other files in this directory is not advisable, because WebLogic Server will try to load any files in this directory, possibly causing server start-up errors.	
CR051215	There is a problem with the password prompt on Solaris. If you press the ctrl-c keys when prompted for the password, the shell window becomes unusable because everything will be written in invisible text.	
CR061030	WebLogic parses the grant statements in deployment descriptors for EJBs, servlets, and resource adapters but does not expand system variables. **Workaround:** Do not use variables in the `<security-permission>` tag in EJB, servlet, and resource adapter deployment descriptors.	

CR064593	The following command-line arguments used to control the SSL time-to-live and cache size are ignored in WebLogic Server 7.0:
	
	`-Dweblogic.security.SSL.sessionCache.size=`
	sessioncachesize
	
	`-Dweblogic.security.SSL.sessionCache.ttl=`
	sessioncachetimetolive
		7.0 Service Pack 5

CR064837	Clients using SSL now must have a valid license. Please refer to the WebLogic Server security documentation for instructions on specifying the license's location. Also, clients using SSL must perform trust checks on the certificates of servers that connect to it by default. Please refer to the documentation on how to specify trusted Certificate Authorities.	

Change Request Number	Description	Release Fixed

CR066782	Statically loading JCE providers, or using Sun's JSSE implementation can cause interaction problems with SSL and WebLogic Server. You may see these errors:
	
	`<<java.security.NoSuchAlgorithm> Exception:`
	`Algorithm XXXXXX not available>>`
	
	or
	
	`<Unable to initialize the server: Fatal`
	`initialization exception`
	`Throwable: java.lang.ExceptionInInitializerError`
	`java.lang.ExceptionInInitializerError:`
	`java.lang.SecurityException: class`
	`"com.rsa.asn1.b"'s signer information does not match`
	`signer information of other classes in the same`
	`package`
	
	Using Sun's JSSE implementation can cause conflicts with WebLogic's SSL implementation, which causes these errors.
	
	Workarounds: Remove the `jsse.jar` (that Sun's JDK 1.4 provides by default) from your JDK directory to resolve the conflict.
	
	You may also see these errors if you are using JCE providers. Loading the providers dynamically rather than statically defining them in the `java.security` file may resolve the problem. Here's an example of a `java.security` policy:
	
	`security.provider.1=sun.security.provider.Sun`
	`security.provider.2=com.sun.rsajca.Provider`
	
	The sunJCE provider is specifically unloaded by WebLogic Server if it is found to be registered statically. If you require the sunJCE provider, load it dynamically.	

CR067087	The `security-permission-spec` attribute cannot currently be added to your `weblogic-application.xml`, you are limited to using this attribute within `weblogic-ejb-jar.xml` and `weblogic.xml`.	
CR067621	The `TLS_NULL_WITH_NULL_NULL` cipher suite is not supported and will be ignored. A warning message will be logged indicating that the cipher suite is not being used.	

Change Request Number	Description	Release Fixed
CR067814	If you wish to use an LDAP browser to view the embedded LDAP server in WebLogic Server, use the following information `host:`*`localhost`* `port:`*`listen port for WebLogic Server`* `User DN:cn=Admin` `Password:`*`the password defined for the embedded LDAP server on the Domain-->Security-->Embedded LDAP tab.`*	
CR072154	The `MyListener` class is implemented in the `SSLSocketClient.java` file. In the Class Summary table in the SSLClient `Package-Summary.html` file, the class is listed as `SSLSocketClient.MyListener`. The file location is: `SAMPLES_HOME\server\src\examples\security\sslclient`	7.0 SP2
CR073187	There is a known issue with the `TLS_RSA_WITH_DES_CBC_SHA` cipher when your SSL license allows only export level cryptography. This situation causes a server that is configured to use that suite to fail to start SSL when you boot the server. An example of the error you will see upon failure is as follows: `<Apr 2, 2002 7:32:24 AM PST> <Emergency> <WebLogicServer> <000209><ListenThread.run() failed: java.lang.IllegalArgumentException: Export Restriction on suite: TLS_RSA_WITH_DES_CBC_SHA java.lang.IllegalArgumentException: Export Restriction on suite:TLS_RSA_WITH_DES_CBC_SHA at com.certicom.tls.TLSSystem.setEnabledCipherSuites(Unknown Source) at javax.net.ssl.impl.SSLServerSocketImpl.setEnabledCipherSuites(Unknown Source) at weblogic.t3.srvr.SSLListenThread.newServerSocket(SSLListenThread.java:667) at weblogic.t3.srvr.ListenThread.run`  **Workaround:** If your server is configured to use specific cipher suites, to allow your WebLogic Server to start up remove the following from the cipher suites configuration setting:  `TLS_RSA_WITH_DES_CBC_SHA` `SSL_RSA_WITH_DES_CBC_SHA`	

Change Request Number	Description	Release Fixed
CR074716	The Administrator, Operator, Deployer and Monitor roles are only partially functional in this release. Adding someone to those roles will give them access to the appropriate administrative MBean operations but will not give them access to the Administration Console.	
	Workaround: To give a user access to both the MBean operations and the console add them to the Administrators, Operators, Deployers and Monitors groups. Furthermore, if you are running in compatibility mode you should make sure that the system user is in the Administrators group.	
CR075658	There is a known issue regarding access control lists (ACLs) for JDBC connection pools when running in compatibility mode or when using File realm as the security realm. If your `fileRealm.properties` file includes an ACL that grants *any* permission on a connection pool, the shrink and reset actions will be disallowed for that connection pool, even if you include ACLs that explicitly grant shrink and reset permission.	
CR081250	A disabled user account in the LDAP Active Directory Authentication provider still authenticates to WebLogic Server.	
	Workaround:	
	Modify the user filter to only return accounts that do not have the `UF_ACCOUNTDISABLE` bit set. For example:	
	Original Filter `microsoft.user.filter=(&(sAMAccountName=%u)(object class=user))`	
	Modified Text `microsoft.user.filter=(&(sAMAccountName=%u)(object class=user)(!userAccountControl:1.2.840.113556.1.4 .803:=2))))`	
CR082106	If a user account becomes locked and you delete the user account and add another user account with the same name and password, the UserLockout attribute will not be reset.	

Change Request Number	Description	Release Fixed
CR082926	The SimpleConnectionFilter2 sample in the Security Network Connection Filter samples may cause WebLogic Server not to boot. This problem is known to occur on Solaris platforms only, but may occur on others. The sample file is at this location: `SAMPLES_HOME\server\src\examples\security\net` For example, when running the SimpleConnectionFilter2 sample on node sunbird, the following filter rule may cause the server to not boot: temple2 10.61.4.13 7001 deny **Workaround:** If the following filter rule is used, the server may boot properly: sunbird 172.18.130.72 7001 deny	
CR083155	The array of Strings returned from the `getValues()` method in the `weblogic.security.service.ResourceBase` class does not support String array value types. Specifically, the JNDIResource "path" key and the EJBResource's "signature" key always return null. If you are using WebLogic 7.0 (without Service Pack1), you may also get a NullPointerException when there are no non-null values in the array.	
CR083242	After you configure a Realm Adapter Authentication provider in a security realm, WebLogic Server will not boot. **Workaround:** Before you configure the Realm Adapter Authentication provider, create an empty `filerealm.properties` file in the directory from which you boot WebLogic Server. When you configure the Realm Adapter Authentication provider, set the Control Flag to `Optional`. Reboot WebLogic Server.	
CR083273	When the username/password combination for a domain is specified as `weblogic/weblogic`, the Administration Console pre-populates the username and password causing a potential security risk. **Workaround:** Do not use the username/password combination of weblogic/weblogic in production.	
CR085089	When creating a Custom Authorization provider, specifying EJB parameters of the same type in the `getname()` method of the `ContextElement` is broken. Only the first parameter of the type is retrieved. This known problem is fixed in WebLogic Server 7.0 SP02.	

Tools Known Issues

Change Request Number	Description	Release Fixed
CR064893	The following deployment descriptor elements cannot be edited in WebLogic Builder:	
	`ejb-jar.xml` elements:	
	• `cascade-delete (EB/Advanced)`	
	• `weblogic ejb xml`	
	• `transaction-isolation`	
	• `description, display-name, large-icon, small-icon`	
	• `exclude-list`	
	• `unchecked`	
	• `use-caller-identity`	
	`weblogic-cmp.xml` elements:	
	• `enable-dynamic-queries (EB/Advanced)`	
	• `isolation-level (Methods/Transactions)`	
	• `multiple table mapping`	
	• `db-cascade-delete`	
	• `field-group, group-name (*)`	
	• `relationship-caching, caching-name, caching-element`	
	• `validate-db-schema-with`	
	• `database-type`	
CR065250	WebLogic Builder may fail to access a database on a server if the driver classes for the database are not on the client.	
CR066466	WebLogic Builder does not provide the option to cancel operations.	
CR066668	On Linux, WebLogic Builder's file browser opens at root instead of current directory.	
CR068152	The operations Add, Edit, and Delete can be slow in the EJB elements of WebLogic Builder.	8.1
CR068165	In the Add dialog in Resource Links, in the WebLogic Settings, J2EE Links, WebLogic Builder incorrectly offers a selection of application scoped data resources.	
CR068620	Table columns may not resize properly in WebLogic Builder.	8.1

Change Request Number	Description	Release Fixed
CR069568	WebLogic Builder cannot map multiple roles to a method in EJB method permission.	
CR071833, CR072541	WebLogic Builder support for RARs does not include generation of deployment descriptors.	
CR072700, CR074591	If you use WebLogic Builder's deployment dialog box to undeploy an application, application the dialog will not show the undeployment. This dialog may also report that applications whose deployment has failed are deployed.	
CR073715	WebLogic Builder may not be able to display multiple RDBMS XML files.	7.0 SP2
CR073975	WebLogic Builder incorrectly allows you to add a transaction without specifying a method or a transaction attribute.	7.0 SP3
CR074210	Optimistic concurrency strategy does not work in WebLogic Builder.	8.1
CR074247	In WebLogic Builder, adding a finder with a duplicate method name may update the original finder instead of adding the new one.	7.0 SP2
CR074322, CR074856	In WebLogic Builder, cancelling an edit action may not work.	8.1
CR074566	Finders or CMP values modified through WebLogic Builder's BeanGrid/Edit (like include-update) may not be represented in the individual panels.	8.1
CR074622	In editing a CMP field using the WebLogic Builder's pop-up dialog, if you set column type to OracleCLOB and then click on the CMP tree node, the column type change doesn't show up on the CMP panel.	
CR074634, CR074955, CR074972	Using WebLogic Builder's Relations wizard to edit the relations of a CMP with multiple relations already configured results in an exception.	7.0 SP2
CR074639	When you edit a Finder method name using WebLogic Builder's editing panel, the change doesn't propagate to XML. When you edit using the Finder pop-up dialog, the change does not propagate to XML or to the editing panel.	7.0 SP2
CR074728	WebLogic Builder does not allow changing a method to a local method.	
CR074754	Editing some transaction attributes may cause WebLogic Builder to create duplicate descriptors.	

Change Request Number	Description	Release Fixed
CR074775	It is recommended that you set the enable-call-by-reference element explicitly in WebLogic Builder.	
CR074961	If you use WebLogic Builder to generate web.xml descriptors, it creates a security-role "guest" which pertains to the pre-7.0 security protocol.	
CR075005	Specifying an invalid drive when opening a component (type bogusDrive in the textbox, press enter and select open) in WebLogic Builder causes a NullPointerException.	7.0SP2
CR075006	In some circumstances WebLogic Builder will not properly construct one-to-many relations.	7.0 SP2
CR075129	If you switch a stateless session bean to stateful, WebLogic Builder may not show that the module is dirty.	
CR075228	WebLogic Builder may throw an assertion error if you delete a transaction or permission method.	8.1
CR075586	If you want to use WebLogic Builder with the WebLogic Server Examples Server or examples, you should run setExamplesEnv (located at%SAMPLES_HOME%\server\config\examples) before starting WebLogic Builder, so that files needed to run the PointBase database server will be added to the classpath. If you do not, you will get database error messages.	
CR075753	Some descriptor elements may pass through WebLogic Builder validation and then be found to be in error by ejbc.	
CR075924	In some cases, WebLogic Builder may not properly generate descriptor files for nested WAR files in EARs.	8.1
CR075936	WebLogic Builder may generate an illegal value for the element <estimated-bean-size>.	8.1

WebLogic Tuxedo Connector Known Issues

Change Request Number	Description	Release Fixed
CR067275	Load balancing and fail-over are not supported with WebLogic Tuxedo Connector in a cluster using RMI over IIOP.	
CR079630	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 to the length difference.	

Workaround: 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 "1234567890ABCDEFGHIJKLMNOPQRSTUV":

```
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.

Web Services Known Issues

Change Request Number	Description	Release Fixed
CR067190	WebLogic Web services do not support XML Schemas that contain complexTypes and elements with the same name.	

CR072890	The command line version of the `servicegen` Ant task does not support the following options that correspond to the Ant task attributes:
	
	```\n-overwrite\n-contextURI\n-includeEJBs\n-excludeEJBs\n-generateTypes\n```
	
	The command line version of the `clientgen` Ant task doe not support the following options that correspond to the Ant task attributes:
	
	```\n-autotype\n-overwrite\n```
		8.1

CR073661	WebLogic Web services that are described with an invalid `web-services.xml` file deploy incorrectly on WebLogic Server, although they are not accessible by client applications trying to invoke them.	
CR073899	When generating a SOAP fault, WebLogic Web services adds extra non-CDATA characters.	7.0 SP2

Change Request Number	Description	Release Fixed
CR083493	The messages and multicomponent examples in the Web Services examples package do not contain the code needed to generate a downloadable client.jar file. This omission does not affect the functionality of the examples that work with the provided client.	
CR101160	Complex data types in service-specific exceptions are not handled in strict accordance with the Java API for XML-Based RPC (JAX-RPC) 1.0 specification, as follows:	

For CR101160:

- **Section 4.3.6 WSDL Fault**, *Service Specific Exception*: When the `clientgen` Ant task generates the JAX-RPC stubs from a WSDL that contains a `wsdl:fault` element whose message part is an `xsd:complexType`, the generated stubs incorrectly map the `xsd:complexType` to a simple Java type. This means that when a client application uses the stubs to invoke the Web Service, and the service throws the service-specific exception (rather than the a `javax.xml.rpc.SOAPFaultException`), the client application cannot access the full complex data from the exception.

  There is no workaround to this problem.

- **Section 5.5.5 Methods**, *Exceptions*, fourth bullet: The WebLogic Web Service runtime incorrectly maps multiple fields in a service-specific Java exception class to a simple XML data type, rather than to an `xsd:complexType`.

  Workaround: always throw a `javax.xml.rpc.SOAPFaultException`, rather than a service-specific exception, when implementing a WebLogic Web Service.

XML Known Issues

Change Request Number	Description	Release Fixed
CR070547	When using the XML Streaming API on Solaris, some character references do not resolve to the same character as on NT.	

Known Issues in WebLogic Server 7.0

This section lists problems in WebLogic Server 7.0 that are not associated with a specific Service Pack.

Connector Known Issues

Change Request Number	Description	Release Fixed
CR098342	The WebLogic Server 7.0 JCA implementation is not compatible with the IBM CICS resource adapter. BEA's implementation of the `javax.resource.spi.ConnectionManager` interface does not direct return the object returned by the `getConnection()` method of the resource adapter's implementation of the `javax.resource.spi.ManagedConnection` interface. It returns an instance of a class dynamically generated by the `java.lang.reflect.Proxy` class instead.	

Core Server Known Issues

Change Request Number	Description	Release Fixed
CR106616	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.	

EJB Known Issues

Change Request Number	Description	Release Fixed
CR061938	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 ( wl1.bal = 300 AND wl0.acct_id=wl1.acct_id );
CUST_NAME   CUST_INTER   ACCT_ID
----------  ----------   ----------
100         jazz
100         jazz
100         jazz
100         jazz
100         jazz
```

JDBC Known Issues

Change Request Number	Description	Release Fixed
CR101419	Fail-over for DataSources and TxDataSources in a cluster is not working in this release.	8.1 SP1

RMI Known Issues

Change Request Number	Description	Release Fixed

CR099693	In 7.0 (G.A., SP01, and SP02), when a client accesses WebLogic Server through a firewall using the t3 protocol, this server error results:
	
	`####<Feb 17, 2003 10:15:59 AM EST> <Error> <RJVM> <ewibeasbx01.merck.com> <edmServer> <ExecuteThread: '14' for queue: 'default'> <kernel identity> <> <000506> <Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection:`
	
	`'JVMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,7002,7001,7002,-1]:edm:edmServer' to: '0S:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0',flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''>`
	
	The server's `ExternalDNSName` was specified. Upon initial connection, the client addressed the request to the server instance's `ExternalDNSName`. The JVMID in the response from the server instance required the server's internal IP address and internal DNS name. Subsequent requests from the client were not recognized.
	
	This is a known problem in WebLogic Server 7.0 that is resolved in SP03. The problem occurs if clients are connecting over t3 through a firewall.
	
	Workaround: add an entry to the external DNS servers that maps the server instance's internal DNS name to its external IP address.
		7.0 SP03

Spaces in Directory Names

WebLogic Server will not start from a directory whose name or address includes a space. For example, the following directory names cause failures on startup:

```
D:\bea new\weblogic700 space\
```

```
D:\bea new\
```

Do not include spaces in directory names in WebLogic Server 7.0. This issue is represented in change request CR084568.

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`.

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.

Clusters, DNS, and Multihoming on NT

Be aware of naming issues when using multihoming features on Windows NT in a cluster. For example, naming conflicts may occur 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.

WebLogic Keystore Provider Support

Only Java 2 Enterprise Edition (JKS) keystores can be used with the WebLogic Keystore provider. JKS keystores that are password-protected can now be used with the WebLogic Keystore provider.

Note: The WebLogic Keystore provider is deprecated in this release of WebLogic Server.

Transaction Limitations with a Non-XA JDBC Driver in a Multi-Server Environment

The Emulate Two-Phase Commit (or `EnableTwoPhaseCommit=true`) option on a TxDataSource is designed to allow a single non-XA resource to participate in a distributed transaction. You can use only one such TxDataSource in a distributed transaction.

In a multi-server environment (including clusters), if you configure a TxDataSource with a non-XA resource on more than one server, an application must access the resource *on the server that acts as the transaction coordinator.* If not, an error will occur during commit processing which will result in a heuristic completion.

In a clustered environment, there is a greater risk for this error to occur due to load balancing between multiple servers.

Workaround:

Use a JDBC driver that supports XA to access JDBC resources in distributed transactions.

For more information about configuring a TxDatasource with a non-XA JDBC driver, see "Configuring Non-XA JDBC Drivers for Distributed Transactions" in the *Administration Guide* at `http://e-docs.bea.com/wls/docs70/adminguide/jdbc.html#confignonXA`

Resolved Problems for Service Pack 7

Service Packs are cumulative. Service Pack 7 contains all the fixes made in earlier Service Packs released for WebLogic Server 7.0. For a description of fixes made in earlier Service Packs, see Chapter 4, "Resolved Problems for Service Packs 1 - 6."

The following sections describe problems that were resolved in WebLogic Server 7.0 Service Pack 7:

- "Administration Console" on page 3-2

- "Clusters" on page 3-4

- "Core Server" on page 3-4

- "Deployment" on page 3-5

- "EJB" on page 3-6

- "JDBC" on page 3-7

- "JMS" on page 3-8

- "JSP" on page 3-9

- "JTA" on page 3-10

- "JVM" on page 3-11

- "Node Manager" on page 3-12

- "Operations, Administration, and Management" on page 3-13

Administration Console

Change Request Number	Description
CR082722	The Deployment Descriptor Editor did not properly handle the `<unchecked/>` method permissions in the `ejb-jar.xml` file. This problem has been resolved.
CR108255	If you tried to create a Japanese user name in the English version of WebLogic Server Administration Console (Group Tab), a `weblogic.management.utils.NotFoundException` was thrown. This has now been resolved.
CR109110	Prior to this release, if the WebLogic Server Administration Console was used to delete a WTC object that had already been deleted, it threw a `InstanceNotFoundException` and displayed the stack trace in the Console. A meaningful error message is now displayed in place of the stack trace.
CR122389	WebLogic Server ignored case sensitivity when it came to names of a node. As a result, it listed a node only once in the JNDI tree even if there were multiple nodes that had the same name, but different case. This problem has been resolved.
CR209949	When comma separated multiple directories were specified as values for the `alternateTypesDirectory` property, WebLogic Server did not load valid archives present in these directories. Archives were loaded only when a single directory was specified for the property. This problem has been resolved.

Change Request Number	Description
CR216670	When you exceeded the number of authentication attempts, a message was logged on the domain displaying that you were locked out.
	Now, the Administration Console also displays the lockout message.
CR216851	The Define Security Policy and the Define Scoped Role pop-up menu options are no longer available for JMS bridge destinations.
CR238054	Monitoring statistics related to EJBs were not displayed in the WebLogic Server Administration Console.
	This problem has been resolved.
CR239018	`CreateException` occurred when you configured a security policy for a JMS destination.
	This problem has been resolved.
CR241554	When the `MSI File Replication Enabled` check box was selected for starting the Managed Server in the Independence (MSI) mode, the configuration file (`config.xml`) was incorrectly copied as `msi-config.xml` instead of `config.xml` in the Managed Server's root directory. Since the `config.xml` file could not be located during server bootup, the Managed Server failed to start in the MSI mode.
	This problem has been resolved.
CR241861	Monitoring statistics related to web application were not displayed in the WebLogic Server Administration Console.
	This problem has been resolved.
CR253611	A `NullPointerException` was thrown in the WebLogic Server Administration Console when you tried to configure a new user after changing the user password.
	This problem has been resolved.
CR255254	The `AuthCookieEnabled` attribute associated with a Web Server MBean could not be set through the WebLogic Server Administration Console.
	This problem has been resolved.

Clusters

Change Request Number	Description
CR241267	If a server was no longer available in the cluster, there was a possibility of an invalid server lookup occuring. This was because an internal hash table was not being properly updated.
	This problem has been resolved.

Core Server

Change Request Number	Description
CR132806, CR126374	The `Thread.currentThread().getPriorty()` method, used to determine the priority of the current thread, returned the default value instead of the actual priority that was set during configuration.
	This problem has been resolved.
CR196369	Refer to the security advisory information available at:
	http://dev2dev.bea.com/pub/advisory/138
CR200568, CR128417, CR134918, CR172274	Sometimes server-side sockets remained in an IDLE state when a client issued a reset on the socket before the server could set the `TCP_NODELAY` option. Therefore, these IDLE sockets were not getting cleaned up properly.
	Now, when an exception occurs when setting the `TCP_NODELAY` option on the socket, server-side sockets are cleaned.
CR210488	Exceptions occurred during compilation of an EJB with IIOP on AIX platforms.
	This problem has been resolved.
CR211299	Refer to the security advisory information available at:
	http://dev2dev.bea.com/pub/advisory/156
CR218491, CR201647	WebLogic Server threw `UnmarshalException` with nested `OptionalDataException` after EJB tier was restarted, due to synchronization issues with the `BasicReplicaList` class.
	This problem has been resolved.

Change Request Number	Description
CR236296	A memory leak occurred when the `call-by-reference-enabled` property was set to `False` and a JSP was calling an EJB.
	This problem has been resolved.
CR239324, CR239819	If a socket is unregistered and then registered, and if between the two calls the timeout timer never ran, the data structures would get unsynchronized, preventing the newly-registered socket from timing out.
	This problem has been resolved.
CR243895	While invoking a remote EJB, the `ConcurrentModificationException` occurred because a HashMap was simultaneously modified by two threads.
	Now, access to the HashMap has been synchronized.
CR248250	When a remote object was accessed remotely and locally (within the server), the local access caused data corruption, causing load balancing to fail in both cases.
	This problem has been resolved.

Deployment

Change Request Number	Description
CR215116	By default, if auto deployment is enabled, applications are automatically undeployed when they are deleted from the applications directory of the server. However, if an attempt was made to copy an application to this directory (after the delete operation), WebLogic Server would throw an `UndeclaredThrowableException`.
	This problem has been resolved.

EJB

Change Request Number	Description
CR122378, CR200204	When an EJB that had both local and remote interfaces was deployed and accessed from a Web client deployed in the same application (EAR), a `ClassCastException` was thrown when create was called on `LocalHome`. This problem has been resolved.
CR187691	You might encounter a `CacheFullException` if the `findByPrimaryKey` method is called, when the `concurrency-strategy` property is set to `Exclusive`, and the `cache-between-transactions` property is set to `True` for an entity bean. This problem has been resolved.
CR221686	When a JSP with 100 or more EJBs in an EAR file was accessed, compilation by the container failed because of the command length limitation (for the forked process). Now, by default, JSPs and EJBs are compiled inline because inline compilation does not have this limitation.
CR229290	During EJB compilation, Weblogic Server did not honor the options specified in the `extra-rmic-options` element of the `config.xml` file. This problem has been resolved.
CR240649, CR174593	WebLogic Server threw a `ConcurrentModificationException` when a finder of an entity bean was invoked in the `ejbStore` method, and new entity beans were enrolled in a transaction of this method, or when entity beans of the same type were modified in this method. This problem has been resolved.
CR258768	`AssertionError` was encountered when a user with insufficient privileges called the `remove()` method on a stateful session bean. The right message, `AccessException` is now thrown.
CR261092	WebLogic Server no longer throws an `XAER_PROTO` error with MQSeries as the JMS provider.

JDBC

Change Request Number	Description
CR108103	Statistics related to the JDBC connection pool waiters sometimes showed an improper value, under load. This problem has been resolved.
CR214723	`SQLException` occurred when a Portal application was deployed using a 10g driver with Oracle database whose character set was set to JIS16EUC. This problem has been resolved.
CR241211	`NullPointerException` occurred at startup because `XAConnection` created before `TxDataSource` did not have any non-null data source assigned to it when called by `Connection.reserve`. This problem has been resolved.
CR242502	Now, `XAER_NOTA` exceptions that are thrown because of recovery are printed in the server logs only when the `JDBC` debug flag is turned on.
CR249009, CR243824	When enlist failed, `XA.end` was not called, to disassociate the transaction from the connection, before releasing the connection to the pool. This problem has been resolved.
CR256316	`InvalidTransactionException` occurred when a suspended transaction that had timed out was resumed. This problem has been resolved.
CR263367	JConnect Driver for Sybase has now been updated to JConnect 6.05 in the product package.

JMS

Change Request Number	Description
CR172511, CR174691	Closing a `JMSConnection` session before closing all `JMSConsumer` clients first could cause a memory leak. This problem has been resolved.

CR176366	When the `getObject()` method is set on the `Boolean.TYPE` in the `onMessage()` API, it throws the following JMSException:
	
	`javax.ejb.EJBException: nested exception`
	
	`is:weblogic.jms.common.JMSException:`
	
	`Error deserializing object weblogic.jms.common.JMSException:`
	
	`Error deserializing object`
	
	This problem has been resolved.

CR177558	A server-side memory leak no longer occurs with NO_ACKNOWLEDGE delivery mode.
CR178775	When a JMS client repeatedly opened and closed JMS connections against WebLogic Server, `JMSConnection` and related objects such as `DispatcherWrapper` were not being released in both the client and the server side. This caused an `OutOfMemoryError`, which occurred when the client retained one open JMS connection while opening and closing other JMS connections. This problem has been resolved.
CR187610	JMS Bridge was not using the Messaging Bridge Thread Pool for bridge dispatch requests. Now, JMS Bridge uses the Messaging Bridge Thread Pool if one has been configured. If there are not enough available threads configured in the Messaging Bridge Thread Pool, the bridge uses the default execute thread pool. Bridge also logs a warning message if the configured Messaging Bridge Thread Pool size is insufficient and then attempts to obtain a thread from the default execute pool instead.
CR216063, CR256579, CR234758	`MessagingBridgeRuntime` mbean `getState()` is now properly refreshing state information.
CR226919	A JMS memory leak that was caused by a temporary destination `delete()` failure no longer occurs.

Change Request Number	Description
CR229495	Client no longer hangs while waiting for acknowledge completion when the backend JMS server goes down.
CR235404	Memory leaks were encountered if the producer was not explicitly closed before closing the client connection. This problem has been fixed.
CR239642	`NullPointerException` no longer occurs in `BESession.acknowledge()`.

JSP

Change Request Number	Description
CR217352	When a JSP was accessed using an invalid URL, the 404 error was not thrown due to improper JSP mapping. This JSP mapping issue has now been resolved by changing the default value of `exactMapping` to `true` in the code. If required, this value can be set to false in the `weblogic.xml` file.

CR219805	Compilation of JSP no longer fails when Java comments span multiple scriptlets as shown below.
	
	`<%/*%>`
	`[HTML code]`
	`<%*/`
	`{`
	` /*comment*/`
	`}`
	`%>`

Change Request Number	Description
CR242858	When the `wl:summary` tag available in `Weblogic-vtags.jar` was used by JSPs, the query string was not passed to the next page.
	This problem has been resolved by adding the `AddQueryParams` optional boolean field to the `wl:summary` tag. When the `AddQueryParams` field is set to `True`, the query string is appended to the URI.
CR254360	`ClassNotFoundException` was thrown for classes in the `WEB-INF\lib` directory when `weblogic.jspc` was used, and the application code used the context classloader.
	WebLogic Server no longer throws a `ClassNotFoundException` under these circumstances.

JTA

Change Request Number	Description
CR048979	A race condition no longer occurs when multiple enlistments of JTS connections attempt to initialize the resource at the same time.
	As a result, WebLogic Server no longer throws a `javax.management.InstanceAlreadyExistsException` while trying to get a JTS connection.
CR213034	Redelivery of JMS message did not occur after migration on transacted messages.
	This problem has been resolved.

Change Request Number	Description
CR215559	When a resource involving a global transaction threw a Heuristic Hazard Exception, the transaction was declared completed regardless of the fact that all the subcoordinators and resources had not responded to the commit.
	Ideally, during a commit process, if one of the subcoordinating servers is down, the coordinator should not have committed till such time that it heard from all resources and subcoordinators. Because WebLogic Server did not allow all subcoordinators to complete the commit call, transactions remained pending even though the transaction had been declared committed.
	The problem has been resolved.
CR249090	When a transaction reaches the prepare phase, as it approaches its timeout, there is a possibility that a negative value could be passed to the `Object.wait()` method. This results in an `IlegalArgumentException` being thrown by the wait method which is then wrapped by a `SystemException` exception and thrown to the commit caller.
	This problem has been resolved.

JVM

Change Request Number	Description
CR218538, CR241553	A `JAVA LEVEL DEADLOCK` no longer occurs when the RJVM is being shut down.
CR252304	When the RMI service of a remote server was called from a servlet after a `MaxMessageSizeExceededException` occurred, some threads stopped indefinitely at the status of waiting for an RMI response.
	This problem has been resolved.

Node Manager

Change Request Number	Description

CR111339, CR237367	Attempts to start multiple WebLogic Server instances using Node Manager on slower hardware produced the following error:
	
	`<Jun 25, 2003 5:16:15 PM CDT> <Error>`
	`<NodeManager@192.168.190.4:5555>`
	`<__COMMAND_EXCEPTION__Request: failed to execute command`
	`'online' on server 'ifgw1catapp4 - reason:`
	`'[JavaProcessControlOnline: Could not get valid pid for`
	`WebLogic process.]'>`
	
	This problem occurred because Node Manager looked for the PID file before there had been a chance to write to that file.
	
	A new property allows users to specify how many times Node Manager will retry reading the PID from the file. Every retry will have a 2 second wait. The property introduced is `PIDFileReadRetryCount`, which can be specified as a system property or in the `nodemanager.properties` file. When the value of `PIDFileReadRetryCount` is greater than zero, Node Manager attempts `PIDFileReadRetryCount` number of times before giving up on reading the PID from the file. The default value of `PIDFileReadRetryCount` is zero, so there is no change in default behavior.

Operations, Administration, and Management

Change Request Number	Description
CR179126	An `InvalidAttributeValueException` was thrown by the `weblogic.Admin` utility when the `addTarget` method was invoked using the utility on an Administration server hosting a JMS server, and the JMS server was not yet targeted. This problem has been resolved.
CR189763	When a staged application was deployed to a server running in MSI mode, the server incorrectly created a staging directory. Now, when a staged application is deployed on a server running in MSI mode, WebLogic Server determines whether to use an existing staging directory or create a new one. As a result, there is no `InvalidAttributeValueException` when deploying to a server running in MSI mode where the server process does not have write access to the directory structure.
CR196774, CR135780	When a WebLogic configuration property was used while booting WebLogic Server and that property was unknown to the WebLogic Server management subsystem, a message of severity WARNING was being logged. The Message severity was lowered to INFO and the message text was changed to accurately reflect the nature of the message.
CR224350	When there was an unhandled exception in the `handleNotification()` method of `weblogic.management.timer.Timer`, delivery of timer notifications did not take place.
CR235443	The `comm` Mbean Proxy stub that is initialized with the Admin MBean Server stub is now refreshed when the Administration server is rebooted.
CR259326	Passwords in the `Properties` attribute of the `JDBCDataSourceFactory` were not encrypted. Now, a `Password` attribute has been added to `JDBCDataSourceFactoryMBean`. The value specified in this attribute is encrypted and access to the password is restricted to only Administrators. In addition, the `JDBCDataSourceFactoryMBean.Properties` attribute no longer contains the password.
CR262604	`weblogic.Admin PING` can no longer be used by anonymous users. This is acceptable behavior. However, an incorrect error message was displayed. The error message has now been modified to state that anonymous JNDI lookup is not allowed.

Plug-ins

Change Request Number	Description
CR201397	With `Idempotent=OFF`, the IIS proxy plug-in does not retry if it gets a `READ_TIMEOUT` exception after `WLIOTimeoutSecs` has elapsed.
CR205132	Now, Apache 2.0.49 (with mod_wl_20.so) child processes are always terminated on HP-UX.
CR206043	When performing HTTP GET requests that were larger than 4096 bytes, the following error was seen in the `wlproxy` log files: `request too long: XXXX, max is 4096` This problem is resolved and the error is no longer thrown.
CR206340	The failover logic used during WebLogic Server shutdown has been fixed. WebLogic Server no longer returns HTTP 404 after SHUTDOWN sequence is initiated.
CR210417	In a WebLogic Server clustered environment with a plug-in setup, HTTP requests were always routed to SECONDARY during failover. This caused SECONDARY to host all the sessions and utilize more than 50% of the server capacity. Now the plugin does not failover to SECONDARY by default. Instead, it routes the request randomly to the next available server.
CR210497	The `iisproxy128.dll` bits were not available in the installer because the path for the binary in the `build-native.xml` file for the installer was wrong. The path has been rectified and the `iisproxy128.dll` bits are now available in the installer.
CR210899	When the `DefaultFileName` property is set along with the `PathTrim` and `PathPrepend` properties, the final URI is no longer malformed.
CR213809	The older Apache plug-in used an undefined symbol AP_NEED_SET_MUTEX_PERMS on an HP-UX platform. This problem has been resolved. Contact BEA Customer Support for the latest patch.
CR216445	The size limitation for the response header has been removed. The NSAPI plug-in now allows response headers that are greater than 16k.

Change Request Number	Description
CR216621	Changes made by `mod_rewrite` to the URI are reflected only to `request_rec->uri`. Hence, the Apache plug-in now uses `request_rec->uri` by default. Also, a new property, `WLForwardUriUnparsed`, has been added. When it is set to ON, the Apache plug-in uses `request_rec->unparsed_uri` instead of `request_rec->uri`. **Note:** The Apache plug-in with `WLForwardUriUnparsed` set to ON does not work correctly with `mod_rewrite`.
CR216841	Absolute URL was not converted to relative URL before processing it for `pathPrepend` and `pathTrim` when using an unparsed URI. Now, if the Apache plug-in detects an absolute URL, it first converts it to a relative URI before applying the `pathTrim` and/or the `pathPrepend` properties.
CR218494	WebLogic Server no longer throws an error during graceful shutdown using a memory-based locking mechanism on Linux and HP_UX.
CR220685	WebLogic Server no longer retries using a new connection when the following are all true: • Idempotent is set to OFF. • A recycled connection is being used. • The backend server is not responsive.
CR222702	Apache Web server no longer generates huge log files under load if APLOG_NOTICE log level is used within the plug-in.
CR222855	Now, the latest proxy plug-in for SunOne 6.1 serves the `ErrorPage` before `WLIOTimeoutSecs`.
CR223636	In a multiple cluster environment, it was difficult to track all the log messages for a particular request in the plug-in log file. Now, a unique request ID has been associated with each log message.
CR224513	NSAPI plug-in now blocks requests that match the directory names specified as values for the `WLExcludePathOrMimeType` property.
CR229338	Now, the plug-in infrastructure logs errors when it fails to open the default log file and when it fails to resolve the hostname.
CR229990	To avoid a delay of 200 milliseconds when receiving the response from the ISAPI plug-in, the Nagle algorithm has been turned off.

Change Request Number	Description
CR231085, CR205852	This problem has been resolved by removing the transfer-encoding header from the response and the web container, where the `HTTPClusterServlet` is deployed, now decides if the transfer-encoding header should be sent to the client or not.
CR232588	The `CookieName` parameter in the WebLogic Apache module overrides the `CookieName` parameter in Apache `mod_usertrack` module. Replace the `CookieName` parameter with `WLCookieName` for a WebLogic Apache module.
CR233924	Now, HTTPClusterServlet first tries the preferred PRIMARY and then tries the SECONDARY server. If both these servers are unavailable, the request is directed to the next available server.
CR235434	When the NSAPI plug-in with Sun One 6.1 is used, the HTTP response header no longer records two instances of the "Date" field.
CR237058	The W3SVC IIS application pool that ran `iisproxy.dll` sometimes crashed. This problem has been resolved.
CR237675	When a resource hosted on IIS was directly requested, the response was sent in one packet. However, when a resource hosted on WebLogic server was requested via the IIS plug-in, the response generated by the server was sent in two packets, although the server itself sends it as one packet to IIS. IIS Proxy Plug-in now sends the response as one TCP packet including headers and body (as sent by the WebLogic Server to IIS).
CR238106	In a clustered environment, WebLogic Server no longer throws a `java.net.SocketTimeoutException` when a client with WebDAV implementation accesses the Managed Server through the `HTTPClusterServlet`.
CR240593	Graceful shutdown of a cluster node no longer causes in-flight sessions to fail.
CR243059	When the backend WebLogic cluster was on high load, the NSAPI plug-in caused the Web server process to crash. This problem has been resolved.
CR245461	When the Welcome page of a web application was accessed from an Apache Plug-in that had the `PathPrepend` and `PathTrim` properties set, an endless loop of HTTP requests were sent from the browser via the plug-in to WebLogic Server. This problem has been resolved.
CR251562	Apache plug-in no longer fails when client sends large POST data as HTTP chunked transfer.

Change Request Number	Description
CR253814	Apache plug-in no longer fails when HTTP transfer encoding is chunked.
CR255498	The WebLogic plug-in for Apache (for versions prior to Apache 2.0.48) has now been redesigned to create mutexes only if MPM is multi-threaded.
CR259649	`WLExcludePathOrMimeType` property did not work correctly if its value was set to a directory. This problem has been resolved.

RMI

Change Request Number	Description
CR215390	RMI `ClassCastException` no longer occurs when the server sends a request to the client.
CR249240	Java level deadlock no longer occurs in IIOP ConnectionManager when two code paths try to acquire the locks in different order.

Security

Change Request Number	Description
CR136729	Prior to this release, events with severity level greater than or equal to the configured audit severity level were logged.
	Now, the default audit provider configuration has a CUSTOM severity option. If the level is set to CUSTOM and one or more severity levels are specified, only events of the specified severity levels are audited. However, if this option is not set, all events with severity level greater than or equal to the configured audit severity level are logged.
	The system properties that indicate the severity levels are `weblogic.security.auditInformation`, `weblogic.security.auditWarning`, `weblogic.security.auditError`, `weblogic.security.auditSuccess`, and `weblogic.security.auditFailure`.
	For example, to log just INFORMATION and FAILURE events, configure your provider for CUSTOM and set these system properties for WebLogic Server:
	`-Dweblogic.security.auditInformation=true`
	`-Dweblogic.security.auditFailure=true`
CR182006	Certificate verification was failing because the `SubjectAlternativeName` extension to the X.509 certificate was marked as critical.
	WebLogic Server now allows X.509 certificates with this extension marked as critical to be verified during the SSL handshake.
CR210310, CR231457	CA certificates that had the path length constraint field omitted in their basic constraints extension were not recognized as CA certificates. As a result, certificate chains containing such certificates failed certificate validation during SSL connection. Now, such certificates are interpreted correctly as CA certificates with no path length constraint.
CR212348	The `listGroupMembers()` method's SSPI call was aborting with an exception if one of the group member's Distinguished Names was invalid.
	Now, the `listgroupmembers()` method shows the valid groups and ignores the members whose Distinguished Names cannot be validated.
CR215515	Some versions of Internet Explorer hang when WebLogic Server sends 0 length packets.
	Now, you can disable sending 0 length packets by adding the public property, `-Dweblogic.security.SSL.DisableSSLEmptyRecords=true`. Enabling this property can prevent Internet Explorer from hanging due to this cause. The default value of this property is `false`.

Change Request Number	Description
CR215960	JSAFE native library support has been added. Since WebLogic Server 7.0 SP7 ships only the `jsafe.jar` (version 3.4.3), please contact the support service of the vendors for other native libraries.
CR218240, CR231517, CR230119, CR230439, CR233424	Refer to the security advisory information available at: http://dev2dev.bea.com/pub/advisory/122
CR234477	An `InvalidParameterException` was thrown while deploying a resource adaptor when the associated deployment descriptor contained permission statements that used wildcard characters or escaped quotes. This problem has been resolved.
CR242606	Managed Servers no longer fail at startup when the `MasterFirst` flag is set to `True` (as part of the domain-side security configuration), and the default embedded LDAP access control lists are in use.
CR243242	Now, user locking, unlocking and bad password counts behave correctly if the authentication provider is user name case insensitive and `caseSensitiveUserNames` is set to false (the default value) or if the authentication provider is user name case sensitive and `caseSensitiveUserNames` is set to true.
CR250189, CR240904	The `memberURL` attribute of the dynamic group now correctly retrieves the users belonging to the group.
CR250264	Due to the default access control lists shipped with WebLogic Server, users could not change associated passwords. This problem has been resolved.
CR252670	Outbound SSL connection was very slow because `weblogic.jar` was verified for every call made to `javax.crypto.Cipher`. This problem has been resolved.
CR261064	Now, WebLogic Keystore providers can be specified as per the JCA specification and can be configured by an administrator.

Servlets

Change Request Number	Description
CR106364	When a URL that includes a relative path (for example, `/../../test.jsp`) is passed to the `RequestDispatcher.forward ()` method, a `404` error is no longer generated.
CR173426	The `servlet-name` element definition has now been added to the `weblogic700-web-jar.dtd` file.
CR185454, CR110293	The session count was an approximate number but was not initialized back to 0 when the server was restarted. Also the increment/decrement was not synchronized. As a result, a race condition could have contributed to a wrong count. In addition, for replicated sessions the count was not incremented/decremented for `becomePrimary`, `becomeSecondary`, and `becomeUnregistered` callbacks.
	WebLogic Server now obtains the count from the map directly for memory and replicated sessions. For file-based sessions, the count is obtained from the file system and for JDBC, it is obtained via a database query. As a result of this fix, counts will no longer be broken. Note, however, that the runtime MBean returns now two more counts:
	1. Total sessions opened by the server so far.
	2. The maximum open sessions in memory.
	These two counts remain unaffected by this change. They will still use the old method of increment and decrement; they are approximate and represent a number since the time the server was restarted. These two counts will be reset to 0 after the server restarts.
CR204553, CR136735, CR256088	A `SessionData` memory leak occurred in the Servlet Container when using Custom Logger and calling `getRemoteUser`.
	The reference count of a `SessionData` may be increased while logging the request if a custom logger calls the `HttpAccountingInfo.getRemoteUser()`, `getRemoteUse()`, `getRequestedSessionId()`, `getUserPrincipal()` or `isRequestedSessionIdValid()`, because these methods try to acquire the `SessionInternal` object. But the reference count is not decremented.
	The problem has been resolved. With the change, the values returned from `getRemoteUse()`, `getRequestedSessionId()`, `getUserPrincipal()` and `isRequestedSessionIdValid()` of `ServletRequestImpl` are now cached into the `HttpAccountingInfoImpl` object before logging the request, and uses the cached value if a custom logger calls the methods of `HttpAccountingInfo`. This avoids incrementing the reference count.

Change Request Number	Description
CR208696	When the `external_stage` mode was used for deploying a war archive, existing war archives in the `.wlnotdelete` directory were deleted on shutdown and again copied on startup. This problem resulted in performance degradation.
	Now, there is a significant decrease in server boot up time because existing war archives are no longer deleted on shutdown.
CR214104, CR222683, CR228624	SessionID generated with reserved keyword in the first 4 letters was impacting performance. The sessionID generation logic was modified to fix this issue.
CR214457, CR202495, CR236810, CR239392, CR206002	Refer to the security advisory information available at: http://dev2dev.bea.com/pub/advisory/130
CR223609	Web application `ClassLoader` ignored the manifest classpath entry in the application web archive. So, a class not found exception occurred while trying to run the web application.
	This problem has been resolved.
CR229577	In `FormBasedAuthentication`, the requested URL was not getting properly stored, both before and after the user was being authenticated using the form-based login page.
	To store the complete URL (including the http or https protocol scheme) of the original request, the following parameter was added to the `weblogic.xml` file:

	`<container-descriptor>`
	`    <retain-original-url> true </retain-original-url>`
	`</container-descriptor>`

CR236526	During failover between servers, requests containing stale sessions are no longer served in case of JDBC persistence.
CR239831	When a large file was forwarded from a JSP with `autoFlush=false`, an infinite loop occurred in `ChunkOutput.writeStream()`.
	This problem has been resolved.

Simple Network Management Protocol (SNMP)

Change Request Number	Description
CR222114	The enterprise OID for the SNMP `coldStart` trap was incorrect. It has been changed to ".1.3.6.1.4.1.140.625".

Web Services

Change Request Number	Description
CR214959	When an array of elements was populated within a SOAP message, a subsequent element was always passed as a null value to the webservice method variable irrespective of the actual value that was passed in the method.
CR257606	When web service calls were made using SSL, `java.net.SocketException` occurred because the socket that was used to retrieve the WSDL was not closed. This error occurred because there were too many open files. This problem has been resolved.
CR258325	Web service client making an outbound SSL call no longer generates a memory leak.
CR259227	On Red Hat Enterprise Linux 4.0 AS, `generateClientJar` (clientgen API) failed with a `ClientGenException`. This problem has been resolved.
CR262186	`ServiceException` is now thrown when a Service object with no access to WSDL metadata is unable to determine port information.

XML

Change Request Number	Description
CR215016	When doing lookup of DTDs over HTTP from a central Web server, if the URL failed (that is, if the Web server was down momentarily), the XML registry did not renew the DTD in the cache and failed each time the DTD was used. Now, the XML registry renews the DTD in the cache, which resolves the problem.

Resolved Problems for Service Packs 1 - 6

The following sections describe problems resolved in previous Service Packs for WebLogic Server 7.0. Service Packs are cumulative; Service Pack 6 contains all the fixes made in earlier Service Packs released for WebLogic Server 7.0. For a description of problems resolved in the most recent Service Pack, see Chapter 3, "Resolved Problems for Service Pack 7."

- "WebLogic Server 7.0 Service Pack 6 Solutions" on page 4-1

- "WebLogic Server 7.0 Service Pack 5 Solutions" on page 4-49

- "WebLogic Server 7.0 Service Pack 4 Solutions" on page 4-143

- "WebLogic Server 7.0 Service Pack 3 Solutions" on page 4-151

- "WebLogic Server 7.0 Service Pack 2 Solutions" on page 4-260

- "WebLogic Server 7.0 Service Pack 1 Solutions" on page 4-300

WebLogic Server 7.0 Service Pack 6 Solutions

The following sections describe problems that were resolved in WebLogic Server 7.0 Service Pack 6:

- "Administration Console" on page 4-51

- "Clusters" on page 4-53

- "Connectors" on page 4-56

- "Core Server" on page 4-56

Administration Console

Change Request Number	Description
CR097378	A NullPointerException sometimes appeared on the Administration Console while the Web Application Deployment Descriptor was being edited.
	WebLogic Server was not checking for the descriptor being null, which is the case when it can not be parsed. For this reason a NullPointerException was thrown.
	If the descriptor is null, the proper error action will now be called to display the correct message.
CR106965	In the Administration Console, "Monitor All Entity EJBRuntimes..." or "Monitor All Message Driven EJBRuntimes..." incorrectly contained information from multiple EJB JAR files with similar names.
	Now, the Administration Console only displays MBeans associated with a selected scope.
CR132429	The Administration Console had no option for taking input from a user to remember the last entered user. By default, the last entered user was saved as a cookie which was valid for one week.
	Now, the login form on the Administration Console has a checkbox called "Remember my ID on this computer". This checkbox is checked by default. When the checkbox is not checked, WebLogic Server will only remember the user ID for the current session.
CR130110	There was an upload directory problem when the complete path rather than a relative path was specified.
	WebLogic Server now checks for relative and absolute path when uploading.
CR079771	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.
	WebLogic Server now sends a action link which will process the file and send the output to the browser in an HTML format.
CR172233	While accessing the user/group list from the providers, the list appeared blank on the Administration Console if any of the providers was down.
	Now, the Administration Console displays the user/group list for the providers that are accessible and shows an error if any of the providers is down.

Change Request Number	Description
CR174734	Re-creating a deleted realm was causing an InstanceAlreadyExistException. Now, the realm is deleted completely so that the InstanceAlreadyExistException is no longer thrown when a deleted realm is re-created.
CR180635	While uploading an archive file through the Administration Console browser, the archive files were uploaded in the domain directory irrespective of the path specified. Now, WebLogic Server uses the upload directory property while uploading files onto the server so the files are uploaded into the correct directory.
CR181865	The Administration Console was allowing the addition of members into the JMS distributed destination even if their JMS server was not targeted. Now, a member cannot be added to the JMS distributed destination if the member's JMS server is not targeted.
CR096998	When the Administration Console is used to edit the CMP EJB deployment descriptor, WebLogic Server no longer throws a NoSuchMethodException.
CR106399	Whenever a method permission was created from the Administration Console in the examples domain and the server was restarted, WebLogic Server was throwing a page not found exception. Adding `MethodPermission.jsp` for ejb20 has solved the problem.
CR188452	The Administration Console was unable to display the server log. Instead, an exception was being thrown. Code fixes have eliminated the problem. Now, the server log can be displayed in the Administration Console.
CR081329	The Refresh function in the Administration Console has been fixed. The Refresh button no longer opens the page that was previously opened, but instead refreshes the page that is currently opened.

Change Request Number	Description
CR206741	WebLogic Server was making too many RMI calls to find out the deployment status of a component on a target especially if the target was a cluster or a virtual host. In addition, the component status on the target was not depicting the precise deployment status of the target.
	The availability status of a component shows whether it is running on a server or not. It also shows the current availability of a component on all the servers of the targeted cluster or virtual host. The availability status is updated when the targeted servers are shut down either gracefully or forcefully.
	Deployment status of a component is now shown in terms of its aggregated deployment status and availability status. The aggregated deployment status of a component could be Available or Not Available when it is deployed on a server and Available, Not Available, or Partially Available when it is deployed on a cluster or virtual host. The Partially Available deployment status implies that the component is available only on some of the servers of the cluster or the virtual host.
	These changes have minimized the number of RMI calls needed to retrieve the status report for a cluster deployment and have resulted in improved performance.
CR214994	The Administration Console server monitoring versions page now displays the system property, `java.vm.vendor`, correctly.
CR178658	The Servlet Extension Case Sensitive attribute has been added to server and cluster configurations. In addition, the Web App Files Case Insensitive attribute has been added to security domain configuration.
	These attributes specify whether file lookups for Java Server Pages (JSPs) are case sensitive on all platforms except win32; file lookups from standard win32 file systems are always case-insensitive. On case-insensitive file systems other than win32 (such as NT Samba mounts from UNIX or Mac OS that have been installed in case-insensitive mode), specify case insensitive lookups by setting these attributes to false to prevent the JSP from returning its source code. For example, if a JSP is being served from a Samba mount and you have specified case-insensitive lookups, WebLogic Server converts all file name extensions to lower case before looking up the JSP.
	Possible values and usage:
	1. OS (the default value): Weblogic Server relies on the operating system for pattern matching.
	2. True: Weblogic Server enforces case-insensitivity irrespective of the operating system.
	3. False: Weblogic Server enforces case-sensitivity irrespective of the operating system.

Clusters

Change Request Number	Description
CR177776	WebLogic Server encountered distributed dead-locks in a cluster when replicated session http requests landed on a server that acts as neither primary nor secondary for the requests. If multiple such requests landed on servers in a cluster, all the threads in the default thread pool were being exhausted due to this behavior and at some point in time, there were no threads available in default thread pool to receive responses. This lead into distributed dead-lock.
	WebLogic Server no longer deadlocks under these conditions.
CR206375	Load balancing http requests in a cluster sometimes caused the following incorrect error message to be reported:
	`Error while primary becoming secondary,[No old secondary found for roid:<roid>] or [No new primary found for roid:<roid>]`
	This error message indicates a problem with session stickiness configuration either in the plug-in or in the load balancer.
	WebLogic Server now displays a proper message that describes the problem and suggests checking the session stickiness configuration in the load balancer and/or in the plug-in.

Connectors

Change Request Number	Description
CR184893	Classloading from within RAR is no longer slow.

Core Server

Change Request Number	Description
CR108791	When the server state was SHUTDOWN_IN_PROCESS and Runtime.getState was called WebLogic Server was returning the wrong string.
	Now, WebLogic Server returns the correct string which depicts the state of the server.
	If your applications were dependant on the wrong string constant that was being returned, you may need to change the string constant.
CR125000 CR129560	The cluster service was setting the real uid/gid of the process after binding to the multicast port. As a result, errors occurred when WebLogic Server attempted to switch the user after binding to the listen ports.
	Now, WebLogic Server uses the postbind uid/gid during server startup except when binding to ports and only switches to the real uid/gid after listening on all ports.
CR121483	A ConcurrentModificationException was thrown when the monitoring subsystem was trying to read the values of the abbrev table and at the same time the abbrev table was being modified by the rjvm layer.
	Cloning the key set before sending the data to the monitoring subsystem so that the HashMap is not modified simultaneously by two threads eliminates the ConcurrentModificationException.
CR130376	According to the documentation at "Starting and Stopping Servers" if the CLASSPATH is too long, it could be added as a single line to a file and then accessed as -classpath @filename. However, this was not working because when beasvc attempted to load the contents of CLASSPATH file, it sometimes truncated the last character. This only happened when the file did not end with a new line.
	This problem has been resolved.
CR133631	Stopping a windows service configured with beasvc.exe sometimes caused a timeout when a stopclass was specified. The service was timing out because there was a race condition between the stop and main threads of beasvc.exe.
	The race condition has been corrected and the windows service no longer times out.
CR135225	A deadlock was occurring between RJVM and NTSocketMuxer.
	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.

Change Request Number	Description
CR105444	The server would throw a `FileNotFound Exception` if a path for a log file was specified but the directory not exist when using the `-Dweblogic.Stdout` and `-Dweblogic.Stderr` command-line options.
	A code enhancement resolves this issue by creating the directory structure specified by the command-line options when the structure does not exist.
CR099356, CR099005, CR186191	A race condition in JVMSocketManager was causing MuxableSocketDiscriminator.isMessageComplete to throw a NullPointerException which resulted in JVMSocketManager running an infinite loop.
	Now, JVMSocketReader initializes the necessary information in static initializer so that initialization happens only once and there is no more infinite loop.
CR130409	When setting the maximum length of an execute queue with the `-Dweblogic.kernel.allowQueueThrottling` 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.
	This problem was resolved by a code fix to check for the dispatch return value on the caller and use the `sendError()` API to return the 503 response.
CR174955, CR105257	A call to WLECService.getConnectionPoolCount resulted in a NullPointerException even if the call was made before the IIOP Connection Pool was initialized.
	Now, if the IIOPConnection Pool is null, zero is returned. As a result, the NullPointerException is no longer thrown when a call is made to WLECService.getConnectionPoolCount.
CR180426	When WebLogic Server was running in a cluster, a warning message appeared when the DNS name of the cluster was not set. This warning message was degrading the cluster performance.
	Now, the cluster address is always read from the clusterMBean if it is not available via the networkchannel. As a result, the warning message no longer causes performance problems for the cluster.
CR180532	Deadlock sometimes occurred when the domain log and server logs were rotated at the same time.
	Now, WebLogic Server no longer logs messages while rotating the logs. As a result, deadlock no longer occurs during log rotation.

Change Request Number	Description
CR173958, CR185090	Interoperability between 6.1 SP4 and 7.0 SP5 domains using `t3` failed 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. The problem was resolved by using "anonymous" when doing the bootstrap authenticate call.
CR175607	Installing WebLogic Server as a Windows service immediately after uninstalling it sometimes created wrong registry keys, which could lead to startup problems. A code fix ensure that the registry keys are flushed and properly closed.
CR179262	During startup, WebLogic Server was experiencing a deadlock between weblogic.jms.common.DistributedDestinationManager and weblogic.cluster.MemberManager. Synchronization has been reduced in MemberManager and now there is no longer a deadlock.
CR182838	LocalServerRef did not implement the hashCode method which caused multiple entity beans with different PKs to have the same stub. LocalServerRef now correctly implements the hashCode method.
CR185841	When starting up many Managed Servers concurrently, the Administration Server CPU usage was high. WebLogic Server now makes fewer remote calls to ServerMbean.getName(). As a result, Managed Server startup time in a large cluser is faster and the CPU usage on the Administration Server is lower.
CR172366	Messages printed by beasvc.exe to the event log were not readable in Japanese locale. A code fix ensures that English messages are printed for all non-English locales.
CR181986	WebLogic Server running as a service sometimes ran out of memory if it was using a large number of threads. Reducing the reserve stack size used by beasvc.exe and beasvc64.exe from 1mb to 256kb eliminated the memory problem.

Change Request Number	Description
CR182684	Every time the beasvc service handler was called, the beasvc log added a line indicating that it was called.
	For example:
	Tue Apr 27 11:52:17 2004] [I] [service_ctrl] 4
	This caused the log to fill up when there was no real activity on the server.
	The debug statement was removed, eliminating the log problem.
CR188371	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.
	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.
	The runtime descriptors for the clusterable stateless session beans now have the propagate-environment attribute set to true by default.
	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.
CR190417	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.
	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.
	Code was added to WebLogic Server which fixed the problem so that it will not deadlock.
CR190507	Fixed a problem with oneway calls for ReplicationManager.

Change Request Number	Description
CR203523	The following log message has been reinstated to appear as it had in previous WebLogic Server releases. `<Warning> <WebLogicServer> <000333> <Queue usage is greater than QueueLengthThresholdPercent "3%" of the maximum queue size. We'll try to allocate ThreadsIncrease "5" thread(s) to help.>`
CR206459	The maximum limit on the number of threads that could be created on the client side was fixed at 50. Now, the maximum limit on the number of threads that can be created on the client side is set to 65536.

CR183620	The startup script for the Managed Server, startManagedWebLogic.sh or startManagedWebLogic.cmd, that was generated from the domain configuration wizard had the following incorrect line:
	
	`cd @USERDOMAIN_HOME`
	
	This incorrect line caused the following errors:
	
	`C:\bea70sp5\user_projects\mydomain>echo off`
	
	`The system cannot find the path specified.`
	
	To fix the problem, the domain configuration wizard now does string substitution.

Deployment

Change Request Number	Description
CR179645	Now, WebLogic Server unpacks large JAR files much more quickly.
CR183537, CR187458	If no target was specified during the redeployment of an application, weblogic.Deployer was adding the Administration Server as the default target. Now, if no target is specified, the deployed application is correctly redeployed to all of the current targets. A new application is deployed to the Administration Server by default.

Change Request Number	Description
CR192196	A WebLogic Server 8.1 client calling `DeployerRuntime.getDeployerRuntime(MBeanHome)` was getting an InstanceNotFoundException if invoked against a WebLogic 7.0 Administration Server.
	A change was made to the implementation of `DeployerRuntime.getDeployerRuntime(MBeanHome)`. As a result, a WebLogic Server 8.1 client is now able to fetch the DeployerRuntime MBean from the WebLogic 7.0 Administration Server and vice versa without getting any InstanceNotFoundException.
CR124450	If the server is running in development mode and an attempt is made to delete an EAR file deployed on the server using the Microsoft Windows Explorer from within the applications directory, the following error no longer occurs:
	`Cannot delete abc.ear:there has been a sharing violation, The source or destination file may be in use.`

EJB

Change Request Number	Description
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.
CR060229	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.
	The Administration Console now allows monitoring of these items.
CR087261	The EJBDeployer was writing an incorrect deployment message to the log for Message Driven Beans.
	The correct message is now being logged when a Message Driven Bean is deployed.

Change Request Number	Description
CR127369	An AssertionError was sometimes thrown when more than one bean was based on the same Java class.
	This error occurred when the following conditions were satisfied:
	1. a bean A had a many-to-one relationship to a bean X (unidirectional relationship)
	2. a bean B had a many-to-one relationship to a bean X (unidirectional relationship)
	3. beans A and B were two different deployments based on the same java class.
	While processing the relationships of beans, WebLogic Server holds the list of cmr-field names, and if the cmr-field name has not been declared, WebLogic Server creates it based on the bean class name. In the above case, while processing relationships of bean X, the cmr-field names of the relation to bean A and the one to bean B will be created. But these class names are the same, so the created cmr-field names are the same. This causes the AssertionError.
	Code has been added to make the cmr-field name unique, eliminating the possibility of conflicting names.
CR108169	When an EJB deployment descriptor was edited using the Administration Console, an incorrect method name was introduced.
	The correct method name has now been implemented.
CR135410	When the bean is looked up and a business method is called for the first time, the call to EntityContext.getCallerPrincipal() from ejbStore returns the user as "anonymous". After this, when ejbStore is invoked again, it returns the correct username. This problem only occurs with ejbStore and no other method.
	This problem has been resolved.
CR187121	A high value for idleTimeoutSecs, for instance, 60000000, in the Deployment Descriptor when multiplied by 1000 to convert it into msecs 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.
	The variables within the EJB container which held the timeout values in milliseconds, such as idleTimeoutMS, sessionTimeoutMS, and readTimeoutMS, have been changed from the int type to long. This prevents any numeric overflow.
CR189847	Stateful session bean (SFSB) InMemory replication was causing an apparent memory leak.
	Although no memory leak was occurring, the heap memory was not being used efficiently. Now, during SFSB InMemory replication, the heap memory is being used more efficiently.

Change Request Number	Description
CR196593	Servers were hanging during stateful session bean (SFSB) replication due to lock contention in the NRUCache. The lock contention in the NRUCache has been reduced to prevent the servers from hanging during SFSB replication.
CR197817	When WebLogic Server executed a finder method on an entity bean, a NullPointerException was being thrown. To eliminate the NullPointerException, the generated code now expects that the bean could be null and the bean is now handled appropriately.
CR132510	When enabling the relationship caching on an entity bean that is using optimistic concurrency and has cache-between-transaction set to true, WebLogic Server no longer throws an IllegalStateException.
CR205974	When a collection valued CMR field is accessed in a transaction other than the one in which it is created, WebLogic Server no longer throws an IllegalStateException. As a result, WebLogic Server delivers the correct result set from the finder query regardless of whether caching is turned on or off.
CR203644	Replicated Stateful Session Beans in a cluster with InMemory replicated session no longer throw a NoSuchObjectException or a LeasedRemoteRef error after instances have been passivated or after the cluster instance has been shut down.

Installation

Change Request Number	Description
CR121040	It was not possible to change the password of a WebLogic Server Windows Service without having to uninstall and re-install the service. WebLogic Server now has an edit option so that parameters can be edited without having to re-install the service. Individual or multiple options can be edited simultaneously as follows: beasvc -edit -svcname:wls_service -javahome:"C:\java\java142_05" -password:"blah" (this command alters the value of "javahome" and "password" parameters) The options will take effect when the service is restarted.

Internationalization

Change Request Number	Description
CR079432	MessageLocalizer was not setting the l10n_package attribute in localized catalog files using the l10ngen utility.
	MessageLocalizer now correctly sets the l10n_package attribute in a localized catalog file.

J2EE

Change Request Number	Description
CR132360	The weblogic.j2eeclient.Main did not work in webstart. The client jar file is an argument loaded by the webstart client and made available in the classpath. But the file was not available in the current directory resulting in an error.
	A system property (weblogic.j2ee.client.isWebStart) was added to WebLogic Server to load the client jar file from the system classpath. weblogic.j2eeclient.Main now works in webstart.
CR103309	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-70.00.jsp.
CR186016	While redeploying an application, J2EEApplicationContainer is now redeploying the internal modules such as JDBCModule as it did prior to WebLogic Server 7.0 Service Pack 5.
CR097079	When an application is removed from the Managed Server, the auxiliary jars are now also removed from the staging directory of the server.
CR103506	When deploying an exploded .ear directory that contains a directory whose name contains a period, ".", a NullPointerException is no longer being thrown.

JCOM

Change Request Number	Description
CR185934	COM to Java clients failed when the Microsoft Security Bulletin MS04-011 was applied. This has now been resolved.
CR199844	When an exception occurred, COM sockets were leaking. Now, COM sockets are properly closed when an exception occurs.

JDBC

Change Request Number	Description
CR136746	Explicit naming of JDBC driver used in the class VendorId were not possible. The problem has now been resolved.
CR127720	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 `XAER_PROTO` or `XAER_RMERR` when an `xa_start()` 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. This problem has been resolved by a code change in the recovery method that prevents special XA connections from being released to the pool twice.
CR174126	Using multiple different prepared statements in a single transaction invoked a bug in the XA statement cache, which lost the handle to unclosed JDBC statements. This quickly caused all the available cursors in Oracle to be consumed and leaked. A fix was made to the underlying collection object being used to implement the XA statement cache.
CR177220	In a multi-server domain, a client involved in a global transaction got the following exception: `weblogic.transaction.RollbackException: delist() failed on resource jdbcXAPool` The code was changed so that if the data source is null, then the resource manager is not set. When the data source is initialized, the resource manager is wrapped by VendorXAResource.

Change Request Number	Description
CR180097	The JDBC connection pool's "CountOfTestFailuresTillFlush" property neglected to clear statement caches.
	The problem was resolved with a code fix.
	For detailed information about using the "flush pool" feature, see "JDBC Connection Pool Testing Enhancements" in *Programming WebLogic JDBC*.
CR184819	WebLogic Server multipools were not failing over when multiple instances of RAC were configured and the first RAC instance was stopped.
	Multipool now fails over correctly.
CR182410	Under high load conditions, WebLogic Server slowed significantly.
	Removing synchronization in the `weblogic.jdbc.common.internal.MultiPool.searchLoadBalance()` method eliminated the slowdown under high load conditions.
CR189978	Misconfiguration of the pool or an unresponsive database could have resulted in the creation of a pool that was not activated. If the pool was never activated, WebLogic Server threw a ResourceException. The JDBC MultiPool would not fail over to the next active pool when this exception was thrown.
	Now, WebLogic Server throws a ConnectDisabledException with a message indicating that the pool was never activated. The JDBC MultiPool detects and handles this exception and does a retry on a different pool if one is available. As a result, the JDBC MultiPool now fails over to an active pool if any of the pools in the list had never been activated.
CR187604	A getMoreResults and a getResultSet call on a JTS Statement wrapper no longer returns the same and already-closed result set.
CR196738	The MSSQLServer4 driver's Statement.cancel() method was corrupting the JDBC connection. JDBC statements were not being closed within a transaction after the connection was closed.
	The MSSQLServer4 driver's Statement.cancel() method was changed so that it is now a no-op and therefore no longer corrupts the connection. This change has no effect on rolling back transactions.

Change Request Number	Description
CR197163	The JDBC connection pool's testing of connections was consuming a lot of database resources because each test was creating a new plain statement which requires the DBMS to parse and plan the test SQL every time.
	Now, pools will reuse a single prepared statement for a connection's test which results in improved performance. However, if any application DBMS tables or procedures are referred to in the test SQL and if they structurally change at runtime, such as an index being added, this may invalidate the test PreparedStatement's query plan. As a result, the subsequent test will fail and the connection and test statement will be replaced. The test SQL suggested by the Administration Console will typically not include any structurally changing table, so the problem of needlessly recycling a connection is now minimized.
CR203460	When a pool connection was made by the XAConnectionFactory, its seconds-to-trust value was not set, so it remained the default.
	Now, the sectonds-to-trust value is set.
CR193357	JDBC refresh was miscounting the number of unreserved connections.
	JDBC refresh now correctly counts the unreserved connections and, therefore, is no longer doing any unnecessary work.
CR087241	A JDBC application using XA calls and any Oracle driver, was getting an ORA ORA-01002: fetch out of sequence error while fetching rows if a transaction was suspended and resumed.
	The AddOracleFlagToXAResource flag has been added to fix this problem when the XA calls use the Oracle 10g thin driver. To avoid getting the ORA ORA-01002: fetch out of sequence error while using XA calls, you must use the Oracle 10g thin driver and turn on the JDBCConnectionPool flag.

jDriver

Change Request Number	Description
CR142730, CR190124, CR190125	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:
	<Warning> <JDBC> <001096> <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.
	and
	<Warning> <JDBC> <001096> <Refreshing this bad pool connection failed weblogic.common.ResourceException: java.sql.SQLException: LDA pool exhausted - make sure you call Connection.close()
	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).
	The jDriver now cleans up connection creation attempts that fail.
CR136168	WebLogic jDriver may cause the server to crash with Oracle error message ORA-02392 when using the long raw type under heavy loads.
	A code fix was implemented to resolve this issue.
CR172462	The WebLogic Server jDriver was not functioning properly with Oracle 9.2 when using the AL32UTF8 character set.
	This problem was resolved with a code fix.

JMS

Change Request Number	Description
CR099554, CR195519	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. Pending messages are now recovered upon restarting the server.
CR135578	The bytes pending counter for non-durable topics was not getting updated correctly when the consumer exited with outstanding non-durable topic messages still pending Redelivery delay on the topic. Now, the bytes pending counters for JMS statistics will be appropriately adjusted for non-durable topic messages when the consumer exits and Redelivery parameters have been configured. There is no redelivery attempt on unacknowledged non-durable topic messages when the consumer goes away.
CR134155	JMS messages from topic sessions caused a memory leak when the receiver was created using a transacted session, used AUTO-ACKNOWLEDGE mode, and the message was rolled back. A code fix resolved this issue.
CR178775	When a JMS client repeatedly opened and closed JMS connections against WebLogic Server, JMSConnection and related objects such as DispatcherWrapper were not being released in both the client and the server side. This caused an OutOfMemoryError which occurred when the client retained one open JMS connection while opening and closing other JMS connections. A change in the code has fixed the problem.
CR179737	A race condition sometimes prevented JMS from detecting and cleaning up lost connections. The reconnect logic was tightened to prevent this type of race condition from occurring.
CR173565	JMS messages were delivered twice when an XA transaction failed. A code fix was made to handle failed transactions, and to log an appropriate error message to notify the user that there is a transaction in an ambiguous state. Prior to this change, any message involved in a transaction that was in an ambiguous state would still be rolled back and redelivered because the message was in memory. However, the record in the store (if there was one) was not properly updated because the store's handle for that record was invalid, leaving the record (if there was one) in the store. Therefore, when the JMS server was rebooted the message was redelivered, resulting in duplicate message. Now any message involved in a transaction that is in an ambiguous state will require a JMS server restart before it can be recovered. Any attempts to recover the message without JMS server restart will result in an RMERR.

Change Request Number	Description
CR183572	Because the JMS IOThreadPool name was incorrect, all JMS stores on a WebLogic Server were sharing the same two threads. As a result, performance was potentially being degraded when more than two JMS stores were running on the same WebLogic Server.
	The JMS IOThreadPool name is now correct which eliminates the potential performance degradation.
CR178405	During a scan for expired messages, a `java.util.ConcurrentModificationException` was thrown.
	The problem was resolved with a code fix.
CR182338	A receiver stops processing messages when a RedeliveryLimit is configured and the number of times the RedeliveryLimit is reached is greater than the MessagesMaximum setting on the ConnectionFactory.
	Example: RedeliveryLimit=0, MessagesMaximum=10. Receiver receives a message and calls session.recover() 10 times. Receiver will stop processing messages and the console will show 10 messages pending.
	Code was added to adjust the window counter when a message is removed from the queue because the RedeliveryLimit had been reached.
CR188040	WebLogic JMS was not receiving notification when ServerDebugMBean() JMS attributes were changed.
	The ServerDebugMBean() can now be used to dynamically enable and disable "DebugJMS" ServerDebugMBean attributes. This allows customers to dynamically enabled or disable JMS Debugging flags.
CR187945	JMS DistributedDestination load balancer was not recognizing pre-existing consumer(s) when a member was added dynamically.
	WebLogic Server now checks to see if a member has any consumers when a member is added to a distributed destination so that the load balancer has the information it needs to load balance correctly.
CR190438	A Java-level deadlock was found between objects weblogic.jms.backend.BEQueue and weblogic.jms.backend.BEConsummer.
	This problem has been resolved by updating the lock hierarchy to make sure that the destination has been locked down first.

Change Request Number	Description
CR197857	JMS was not sustaining the minimum flow rate. The actual minimum flow rate was approximately double the configured value. When using the default flow control settings on a Connection Factory, FlowMinimum=50, JMS was sustaining a flow minimum of approximately 100 messages per second rather than 50 messages per second.
	Now, once a JMS destination exceeds its specified bytes or messages threshold and flow control is started for the producer, flow control throttles back until it reaches FlowMinimum (default 50) and it should maintain that minimum flow rate as long as thresholds are exceeded.
	This change causes a greater slow down from flow control than previously seen because JMS is now correctly sustaining the minimum flow rate.
	To maintain previous flow control behavior (behavior prior to the release of 8.1 sp4), double any FlowMinimum configurations and add a FlowMinimum=100 on ConnectionFactories that were previously assuming default values.
CR187610	Bridge was not using Messaging Bridge Thread Pool for bridge dispatch requests.
	Now, Bridge uses the Messaging Bridge Thread Pool if one has been configured. If there are not enough available threads configured in the Messaging Bridge Thread Pool, then the bridge will use the default execute thread pool.
	Bridge also logs a warning message if the configured Messaging Bridge Thread Pool size is insufficient and then attempts to obtain a thread from the default execute pool instead.

JNDI

Change Request Number	Description
CR197207	The Java client was hanging if the JNDI lookup was performed from a static block during the response from an EJB call.
	To fix the problem, a new system property (-Dweblogic.rjvm.t3.dispatchOnCompleteMessage) has been added. This property needs to be enabled on the client side. When this property is enabled, the t3 request is processed on a thread other than the socket reader thread.

JSP

Change Request Number	Description
CR087857	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.
	WebLogic Server no longer encodes the binary data when ServletOutputStream.write(int) is called.
CR101992	A web application that had a local EJBObject reference in its session, sometimes got an javax.ejb.EJBException after it was redeployed.
	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.
CR174837	WebLogic Server was not allowing custom PageContext implementations.
	The unnecessary cast to weblogic.servlet.jsp.PageContextImpl in the generated JSP has been removed. As a result, custom PageContext implementations are now usable.
CR172380, CR198902	A struts-based application failed to pre-compile an EAR file, and throwing the following exception:
	`<Error> <Deployer> <149205> <The Slave Deployer failed to initialize the application dos due to error weblogic.management.ManagementException: 149233 - with nested exception: [java.lang.ExceptionInInitializerError]. java.lang.ExceptionInInitializerError: java.lang.NullPointerException`
	A code change resolved the problem by setting the thread class loader before precompiling all the JSPs to the `ServletClassLoader()` method. This ensures that the same classloader gets used for loading the struts classes and loading any other class that the struts library tries to load afterwards.
CR085091	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.
	The logic was fixed to serve the custom error page if one is defined.
CR180425	WebLogic Server was sending wlsproxy specific headers even when the request did not originate with a proxy.
	Code was added to check if the request is coming from a proxy and send the appropriate header.

JTA

Change Request Number	Description
CR130073	When an application issued a number of rollbacks from Tuxedo 7.1 through WTC to WebLogic Server, some Oracle XA connections were leaked and not returned to the pool. XA Debug showed that XA End was not called so the connection was not cleaned up. WebLogic Server now calls XA End, allowing the connections to be returned to the pool and eliminating the out of memory error.
CR125476	When a transaction was in the prepare phase and an XAER_RMERR or an XAER_RMFAIL exception was thrown, the transaction remained pending instead of being rolled back. Now, under these conditions, the transaction is rolled back.
CR184941	After a manual JTA migration to back up the server, attempts to boot the crashed WebLogic Server were not successful. Now, after a manual JTA migration to back up the server, if the crashed WebLogic Server is restarted, it boots successfully.
CR202386	Transaction Manager was failing to invoke xa_start() on an object associated with a new resource manager. To fix this problem, the default registration has been standardized. If the application does not call registerResosource(), the resource is registered with the Transaction Manager as a standard resource.
CR188558	When the WebLogic Server was shut down before the timer thread went off, some resources were not being checkpointed to the TLOG. When the WebLogic Server was brought back up, recovery on those resources would not run and any pending transactions would be left pending. Now, when the transaction enlists with a resource, if the resource is coordinated locally and if the resource needs to be checkpointed, it is flagged so that at prepare time, if the transaction is two-phase, the resource gets checkpointed to the TLOG.
CR173464	When getting the cached coordinator, the thread no longer goes into an early notification state which was causing the server to hang.
CR211194	Calling the XA call sequence, XA.end (TMSUSPEND), followed by XA.end (TMFAIL), followed by an XA.rollback in the case of an MQSeries resource prevents memory leaks in MQSeries.

JVM

Change Request Number	Description
CR132228, CR188670	Harmless IOExceptions were not being suppressed on Japanese locales.
	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`
	On non-english locales, the harmless exceptions are now suppressed.

Node Manager

Change Request Number	Description
CR135917	The NodeManager native code was sometimes responsible for a segmentation violation if fdopen on stderr failed. This problem has been fixed.
CR076968	NodeManager has been updated to consistently send the process termination signal as SIGKILL instead of SIGTERM when the offline method is invoked.
CR128517	Using the Administration Console, if you configure a machine, but do not configure the NodeManager, and then you inquire about the state of the server, the NodeManager is contacted by default.
	If you do not need the NodeManager, set the ListenPort to 0 in the machine configuration. This tells the server that the NodeManager is not configured and it will not try contacting the NodeManager.

Operations, Administration, and Management

Change Request Number	Description
CR132947	When the Administration Server was shut down after the Managed Server in the same domain was shut down, the Administration Server was always throwing a java.rmi.ConnectException.
	This problem has been fixed.
CR187170	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.
	The WebLogic Server now sends AttributeAdd/Remove notifications, in addition to the AttributeChange notification, when an array type attribute values change.
CR108496	When editing an EJB deployment descriptor from the console, the <global/> tag was being changed to <global>true</global> in the deployed EJB jar file.
	An empty tag for getGlobalRole method was added so that validation does not fail after persistence.
CR126191	The DESTROY_POOL command used with the weblogic.Admin tool merely disconnected all the users and suspended the pool. It did not actually delete the pool from the domain configuration as it should.
	DESTROY_POOL now deletes the pool configuration. Use DISABLE_POOL or SUSPEND to suspend the pool.
CR134167	EXISTS_POOL did not work as expected. It was looking for runtime mbeans which may or may not exist even if the pool exists. DELETE_POOL, an undocumented command, was the internal implementation to delete a connection pool; however, this feature was documented as DESTROY_POOL externally.
	The Help menu also displayed some undocumented commands.
	EXISTS_POOL command implementation now looks for configuration mbeans to check whether the pool exists.
	DESTROY_POOL command now uses the underlying DELETE_POOL implementation.
	All the undocumented commands that appeared in the weblogic.Admin help menu are now disabled. Commands include TEST_POOL, REMOVE_POOL, SUSPEND_POOL, SHUTDOWN_POOL, RESUME_POOL, DELETE_POOL. These will not be supported going forward and will not work in 7.0 SP5.
	The Help menu no longer lists these commands.

Change Request Number	Description
CR171906, CR179030, CR208205	The Administration Server was not starting when the Managed Servers were running with an operations user. Now, the operator role allows the user to reboot the Administration Server and recontact the Managed Servers. As a result, the Administration Server can now start when the Managed Servers are running with an operations user.
CR179524, CR203851, CR213673	When the server logs were opened in editors/viewers, while the server was still running, the log froze because the server could not re-open the log file for logging after or during the log rotation. Retry logic has been implemented in the logger so that WebLogic Server now attempts to rotate to an alternate log file when the first attempt fails and continues to log even if all attempts fail.
CR177510 CR135356	Adding or redeploying a module that was part of a large EAR file (600MB) took a very long time. A code change to the WebApp module improved performance: • The WebApp module now considers the update of a module when the delta is for changing the whole module. • To reduce the startup time for large application deployment, deployment was modified to persist the staged targets attribute for configured applications. Remote call traffic between the Administration Server and a Managed Server was reduced during application deployment.
CR182686	A ConcurrentModificationException was sometimes thrown while setting System Properties in a startup class or servlet. WebLogic Server has been modified to synchronize the properties and no longer throws the ConcurrentModificationException when setting System properties.
CR184787	The Administration Console did not display the "replicated_if_clustered" value for PersistentStoreType on the session parameters page of the Deployment Descriptor page. As a result, when the descriptor was persisted, it put in a different value into the descriptor. The Administration Console now displays the value and it is correctly added to the descriptor.
CR187170	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. The WebLogic Server now sends AttributeAdd/Remove notifications, in addition to the AttributeChange notification, when an array type attribute values change.

Change Request Number	Description
CR190237	The WebLogic Server `MBeanHome` method `getAllMBeans()` threw an exception when running in a WebLogic Portal environment.
	WebLogic Server `MBeanHome Helper` needed additional knowledge of WebLogic Portal mbean class locations and a correction to existing logic.
	The `ClassNotFound` exception is no longer thrown in the WebLogic Portal environment.
CR188701	Restarting the Managed Servers when the Administration Server was down caused deployment to fail. When the server was set to stage applications, but the application being deployed did not have its staging mode set, the wrong staging mode was used.
	Now, applications that do not have a staging mode set, but are deployed on a server with the staging mode set, are correctly deployed in the right staging mode.
CR190822	JDBC connection pool cloning was not working.
	Now, JDBC connection pool cloning works even if the server is targeted.
CR200310	The application manager was being started twice when run in MSI mode.
	Now the application manager is only started once when run in MSI mode.
CR135521	When creating or deleting a Commo MBean, WebLogic Server no longer throws an `IllegalArgumentException`.
	Now, WebLogic Server throws an `InstanceNotFoundException` if the type is null. A client such as wlconfig catches this exception and uses JMX MBean Home to get the attribute value on the MBean.
CR182072	Now, the ConfigurationVersion attribute in the config.xml file changes value after the service pack is upgraded.
	During every server bootup, the attribute is now set to be the same as the Administration Server version.
CR100218	Because the value for `ExecuteQueueMBean.setThreadPriority()` is non-configurable by a user, `ExecuteQueueMBean.setThreadPriority()` has been excluded from the public javadocs.

Change Request Number	Description
CR109591	The Administration Console was not locating a log called 'myserver_%yyyy%_%MM%_%dd%_%hh%_%mm%.log.
	To fix this problem, WebLogic Server now attempts to locate the filename in the following format:
	`myserver_yyyy_MM_dd_hh_mm.log`
	instead of the following format:
	`myserver_%yyyy%_%MM%_%dd%_%hh%_%mm%.log`
	If still unable to locate the filename using the myserver_yyyy_MM_dd_hh_mm.log format, WebLogic Server throws a FileNotFoundException.
CR124169	WebLogic Server no longer throws an Assertion error when shutting down or undeploying the MessagingBridge using JMX.
CR202449	When a server instance was created on the Administration Server, its default queue was not getting initialized until the Administration Server was reinitialized.
	Now, the default queue is initialized when the server instance is created or cloned.
CR208697	WebLogic Server decremented the notificationID variable when removeNotification() was called to remove some middle elements. WebLogic Server was then throwing an InstanceNotFoundException if access to the last element of
	allNotification[] was attempted.
	As a result, it was not possible to access the last element because the value of the variable, notificationID, was less than the size of allNotification[].
	Code changes have fixed the problem so that now when removeNotification() is called, an ArrayOutOfBoundException is not thrown, and an InstanceNotFoundException is thrown only if the notification instance is not available. The InstanceNotFoundException is no longer thrown if access to the last element of allNotification[] is attempted.
CR211194	Calling the XA call sequence XA.end (TMSUSPEND) followed by XA.rollback no longer causes a memory leak in MQSeries.
CR211815	The locking sequence used during the process of unregistering an MBean has been corrected to avoid deadlock.

Plug-ins

Change Request Number	Description
CR095984	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. The behavior of MaxPostSize configuration is now the same with or without a plug-in.
CR100070	It was not easy to maintain the WLLogFile for each virtual host in the Apache-WLS plug-in because there was only one global debug flag to enable or disable debugging for all the requests handled by the plugin. Now, it is possible to specify the debugging option at the top most level and overwrite it within each virtual host and/or location tag.
CR136968	Weblogic Server was not accepting more than one header when the response.addHeader method was used. The plug-in now allows WWW-Authenticate to have multiple values.
CR171978	When the `FilterPriorityLevel` was set in the `iisforward.ini` file, the forwarding path was broken. A code fix was implemented to ensure that when a virtual host was not defined in the `iisforward.ini` file, the `iisproxy.ini` file from the same location as where the `iisforward.dll` file was loaded is used.
CR172072	Provided an enhancement to the `WLExcludePathOrMimeType` parameter allowing it to be used at the `Location` tag level. `WLExcludePathOrMimeType` parameter can now be defined locally at the Location tag level as wells as globally. When the property is defined locally, it does not override the global property but defines a union of the two parameters.
CR173581, CR173878	The plug-in was logging a confusing "error page is unavailable" log message to the `apache error.log`, even when the client had closed the connection. This was resolved by a code change that commented out the erroneous log message.
CR174431	The Iplanet plug-in now gracefully handles an EINTR OS error.
CR175672	The Apache server is hanging when the WebLogic plug-in tries to open the `wlproxy` log file, even though Debug is OFF. The code has been fixed so that the log file is not set if debugging is turned off.

Change Request Number	Description
CR175989	The Apache server generated core dumps when using the worker (multi-threaded) option instead of the prefork (only multi-process) option.
	This was resolved by fixing the Locking and Unlocking logic.
CR177707	When using the release 7.0 SP02 plug-in with client certificates, WebLogic Server worked fine. However, after an upgrade to release 7.0 SP05, the server log 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`
	The error occurred because the 7.0 SP06 plug-in truncated the `WL-Proxy-Client-Cert` header when it sent it to the server instance.
	The code was changed so that `WL-Proxy-Client-Cert` is lazily added to the request sent to WebLogic Server.
CR179537	The IIS proxy plug-in caused heap corruption on the Microsoft Windows platform.
	The problem was resolved with an internal code fix.
CR180236	The previous 7.0 release 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.`
	The error occurred because the plug-in truncated the `WL-Proxy-Client-Cert` header when sending it to the WebLogic Server instance.
	The problem was resolved with a code fix.
CR178792	HTTP requests can contain either one of the following headers: Content-Length or Transfer-Encoding
	Requests with a Transfer-Encoding header set to "chunked" were failing with an IO error.
	Code was added to support requests using the Transfer-Encoding header set to "chunked".
CR180560	The plug-in was not printing the socket information (localhost:localport remotehost:remoteport) to the log file when making a new connection to WebLogic Server.
	Log messages with the local hostname and local port number are now added when the plug-in makes a new connection to WebLogic Server.

Change Request Number	Description
CR180417	If a cookie was part of the POST data then plugin would corrupt the post data while extracting the cookie.
	Code was added to fix the cookie extraction from Post Data.
CR180724	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.
	The WLCrossOverEnabled functionality now works correctly in WebLogic Server.
CR182434	Headings passed to rq->srvhdrs were not entirely in lower-case instead of mixed case.
	Content-type, content-length and transfer-encoding headers are now passed to NSAPI entirely in lower case.
CR182971	ServerList was deleted after every DNSRefreshInterval which resulted in a core dump.
	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.
CR183188	The ISAPI plug-in was unable to handle requests with the Transfer-Encoding header set to chunked.
	Functionality was added to enable ISAPI to handle such requests.
CR183311	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.
	WebLogic Server no longer creates timer threads when Apache is being used with a single-thread multi-process module.
CR183390	WebLogic Server was throwing an exception from inside the catch block which sometimes caused iPlanet to fail.
	WebLogic Server no longer throws an exception from the catch block.
CR185089	The IIS plug-in was sending an Http status code of 500 (internal Server Error) when it encountered a WRITE_ERROR_TO_CLIENT exception due to a connection closed by the client.
	The IIS plug-in no longer sends Http status code of 500 when a WRITE_ERROR_TO_CLIENT exception is caught.

Change Request Number	Description
CR185668	When using the Apache plug-in to proxy to multiple clusters using MatchExpressions, the PathTrim attribute was failing to trim off the segment of the url used to direct the request.
	Reimplementing MatchExpressions parsing without using the strtok API corrected the problem.
CR186148	When the Apache plug-in encountered a missing page, it was returning a 500 error, rather than the correct 503 error.
	The plug-in now returns the correct error.
CR186470	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.
	WebLogic Server now recycles the connections if the HTTP status code is 302.
CR187282	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.
	WebLogic Server now returns contentLength as -1 if CTE is on.
CR187577	When using multiple Location tags in a VirtualHost tag, the Apache plug-in generated a strange URL if the requests matched two more Locations.
	The Apache plug-in no longer uses regular expression match, unless specified.
CR187578	When KeepAliveEnabled ON was configured in httpd.conf, KeepAliveSecs defaulted to 20. This default setting could not be changed.
	Code was added to ensure KeepAliveSecs is configurable.
CR188811	WLExcludePathOrMimeType did not work correctly when a request had a query string.
	Requests such as http://webserver:port/weblogic/something.jsp?value=123 were not excluded and requests such as http://webserver:port/weblogic/something.do?name=test.jsp were not forwarded.
	The plug-in now ignores query strings while checking for excludes.
CR189251	Under load, Segmentation errors occurred while retrieving plugin Properties for a virtual host.
	Replacing the strtok API with strchr, as the strtok API is not thread safe, eliminated the errors.

Change Request Number	Description
CR190562	Requests were not retried when the plug-in encountered a broken pipe error on Solaris while sending post data to WebLogic Server.
	WebLogic Server now throws a HALF_OPEN_SOCKET_RETRY exception when sendPostData reports a broken pipe on Solaris
CR189933	The WebLogic Server plug-in was not thread safe. The memory address to SrvrInfo array and its size were being passed around and then could be modified by another thread. If they were then modified by another thread, the first thread could end up accessing invalid memory location which could result in seg fault.
	Now, the WebLogic Server plug-in is thread safe. WebLogic Server makes a read-only copy of SrvrInfo array before passing it around.
CR191552	The deprecated property, MaxSkips, was replaced by MaxSkipTime. This new property was not being used throughout the code. As a result, the parameter, MaxSkipTime, defaulted to a value of 10 that could not be changed.
	MaxSkipTime is now used throughout the code and therefore its value can now be changed.
CR192875	The iPlanet Server was crashing if readPostDataIntoFile() threw a new exception from its try catch block.
	This no longer occurs because readPostDataIntoFile() now returns an exception instead of throwing it if it encounters an error while writing post data to a temp file.
CR193447	The cookie did not contain, |, as the delimiter separating primary and secondary information present. As a result, parseJVMID() always returned the primary server information and ignored the secondary server information.
	The cookie is now tokenized to separate primary and secondary information and then call parseJVMID() for both of the extracted values. Now, parseJVMID() returns both the primary server information and the secondary server information.
CR193985	WebLogic Server was logging "creating timer thread in child" messages as [warn] in the log level.
	Now, these messages are logged as [info] in the log level which better reflects the nature of the message.
CR194141	The Apache plug-in was decoding the URI before passing the request to WebLogic Server even though getRequestURI states that the returned value should not be decoded.
	Now, the Apache plug-in passes the URI to WebLogic Server without decoding it first.

Change Request Number	Description
CR194464	A call to internal initJVMID() was not updating the state of the server from GOOD to BAD if the connection was refused. As a result, time was wasted trying to reach a server that was already down.
	Now, if the server is already marked BAD, initJVMID() will skip it and try the next server. Also, initJVMID() updates the state of the server if the connection is refused.
CR199045	When multiple object tags were configured for various path values using both SSL and non-SSL configuration, the plug-in was unable to switch correctly between SSL and non-SSL requests.
	Replaced a global flag (convertDNSSToIP) with a per request flag (moved inside ConfigInfo structure), so that the plug-in now switches correctly between SSL and non-SSL requests.
CR201736	The WL-Proxy-SSL header is no longer missing when using the ISAPI plug-in.

Change Request Number	Description
CR199446	The Apache plug-in did not support regular expressions in the Location and the LocationMatch tags.
	Now, the Apache plug-in supports POSIX regular expressions.
	The following Location tag is not valid anymore:
	<Location */app/*>
	...
	</Location>
	The tag should be changed to:
	<Location /app/>
	...
	</Location>
	Following are some commonly used entries in Location tags:
	Request matches a certain path:
	/app/*
	should be changed to:
	/app/
	Request starts with a certain path:
	/app/*
	should be changed to:
	^/app/
	Request matches a particular mime type:
	/.jsp
	should be changed to:
	\.jsp
CR199668	Hostname matches were case sensitive, so "HostName" did not match "hostname" when doing a host search.
	Now, the host search for iisforward.ini proxy is no longer case sensitive. So, Hostname matches are not case sensitive.
CR202898	When a HEAD request is sent to an Apache plug-in, the headers in the HttpResponse are no longer missing the Content-Length header and the Content-Type header is no longer corrupted.

Change Request Number	Description
CR205009	Starting the Apache plug-in (revision 158) with root user no longer results in a core dump.
CR209778	Code was fixed to ensure that custom value is used for ConnectRetrySecs as specified in the configuration file.
	Now, ConnectRetrySecs is working correctly with Apache.
CR209383	POST data is no longer getting chunked when FileCaching is set to OFF.
CR207694	Now, when "Expect: continue-100" is found in the request, NSAPI replies with "HTTP/1.1 100 Continue" in the header instead of in the response body.
CR205760	If an INSUFFICIENT_BUFFER error was thrown while reading requests, then new memory was allocated. This memory was being accessed later in the method after it had been freed.
	Now, memory is freed at the end of the routine.
CR199080	Now, when the plug-in fails to connect to WebLogic Server, GETLASTERROR is getting logged.
	Log messages have also been added to log any errors that occur while the plug-in is connecting to WebLogic Server.
CR128730	When HttpClusterServlet was sending a request to PRIMARY using a recycled connection, it did not determine that there was any failure to connect until it was too late to re-connect. As a result, HttpClusterServlet was never failing over to SECONDARY.
	Now, HttpClusterServlet checks to confirm whether the request was successfully sent to PRIMARY. Subsequently, if the request to PRIMARY was not sent successfully and HttpClusterServlet had used a recycled connection, HttpClusterServlet tries to create a new connection to PRIMARY. If the subsequent request also fails with an exception, HttpClusterServlet fails over to SECONDARY.
CR208303	On AIX, the errno variable was not thread safe. To make the errno variable threadsafe, `-D_THREAD_SAFE` was added to the `Makefile.aix`.
CR218680	Now for each request, the HTTP status code is set correctly.

RMI

Change Request Number	Description
CR177353	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.
	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.
	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.
CR188748	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.
	WebLogic Server now ensures it truncates extra underscores.
CR197396	The CPU load on machines no longer increases to 100% while server instances call addIndirection for IIOP Outbound responses to clients calling an EJB with RMI/IIOP.
	As a result, performance is now improved.
CR197824	When WebSphere 5.1 tried to make an EJB call to WebLogic server with a HashMap as data, WebLogic Server was throwing an UNMARSHAL exception.
	To fix this problem, WebLogic Server now properly handles foreign service contexts.

Samples

Change Request Number	Description
CR132509	When using the .NET C# example included with the MedRec application (`%WEBLOGIC_HOME%\samples\server\medrec\src\clients\CSharpClient\bin\ReleaseCSharpClient.exe`), 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.

CR133641	iPlanet users experienced a problem with host name verification and received the following:
	```INFO: Host () doesn't match (), validation failed```
	```ERROR: SSLWrite failed```
	A code fix resolved this issue.

Security

Change Request Number	Description
CR121646	A BAD_CERTIFICATE error was received and the SSL connection was terminated when a client certificate with Extended Key Usage set to critical was sent to WebLogic Server. The problem was resolved by adding support for EnhancedKeyUsage. WebLogic Server can now accept Certificates with Enhanced Keyusage set to critical.
CR126837	There was a problem getting a list of users through the listGroupMembers implementation used by the iPlanetAuthenticator MBean. Specifically, the `listGroupMembers()` method returned an empty cursor. This was resolved by changing the default value of the StaticMemberDNAttribute to "uniquemember".
CR137523	If a user was defined while the File realm was calling the refresh() method, synchronization problems with the user table occurred. The File realm has been improved so that exceptions are no longer thrown when defining a user while the File realm is performing a refresh.

Change Request Number	Description
CR180972	Performance of outbound SSL connections was slowed compared to JSSE because WebLogic Server created a new SSLSocketFactory for each outbound request.
	WebLogic Server now caches the SSLSocketFactory when there is no certificate for the request.
CR182860, CR187047, CR194416, CR214190, CR214191, CR214192	When Web browsers sent plain text through the secure port, SSL misunderstood the message header and waited for more data before finishing the message. This delay caused problems with the SSL handshake. The server was using 100%of its CPU usage trying to complete the SSL handshake. This problem was caused by an arithmetic error on the number of bytes read.
	This arithmetic problem has been solved. Clients that send plaintext messages to the secure port now receive errors earlier and the server will not hang because of 100% CPU utilization.
CR185438	Connections to external LDAP servers were sometimes dropped if a load balancer or firewall was configured between WebLogic Server and the external LDAP Server.
	WebLogic Server now automatically refreshes the connections if they are dropped and user requests complete if the connection is dropped during the request.
CR186916	Two aspects of SubjectUtils.isUserInGroup were problematic.
	• Getting only Principals of a specific class required evaluating each Principal while building a set of Principals, then iterating through them to find if any Principal was in the group being searched for. This problem effectively caused two iterations through the Principals when one could b e used to get all Principals of the Subject and then iterating for the group name.
	• Getting an AuthenticatedSubject from the javax.security.auth.Subject interface and searching for the Subject in a group could not be done unless the Subject had been authenticated. This problem meant additional cycles to authenticate the Subject and determine to which groups the Subject belonged.
	RESOLUTION:
	All Principals are now gotten from the Subject and are iterated through once looking for the specified WLSGroup.
	A second method has been added with an AuthenticatedSubject as the first parameter. Both public isUserInGroup () methods were changed to call an internal isUserInGroup ()method that works with a s ate of Principals rather than any Subject. This method eliminates the need for conversion from a Subject to an AuthenticatedSubject.
CR136414	WebLogic Server was unable to negotiate an SSL Handshake with an HTTPS URL connection to a site using the VeriSign Certificate.
	This problem has been fixed.

Change Request Number	Description
CR161839	Synchronization in the TTL cache was not being handled properly thus causing an ArrayIndexOutofbounds exception. Adding synchronization in the TTL cache resolved the issue.
CR177647	In certain cases, WebLogic Server was using its server certificate as a client certificate for the purpose of establishing two-way SSL communication. A change in the code has fixed the problem.
CR186439	When a domain with a cluster is started through the Administration Console, the locking mechanism for the LDAP server is now working properly. As a result, Managed Servers are no longer hanging during startup.
CR189238	When ServletAuthentication.weak was used for authentication, WebLogic Server threw a NoSuchElement exception when the username and password were entered. To resolve the issue, InvalidLogin was modified so that the size is checked before elements are returned.
CR194315	A refresh of the File realm could potentially cause a NullPointer exception. The code modifications have eliminated this issue. NullPointer exceptions no longer occur when the File realm is refreshed.
CR178854	Some of the demonstration certificates and trusted CA certificates shipped in previous service packs of WebLogic Server 7.0 expired on May 14, 2004, or will not work with the Basic Constraints feature. WebLogic Server 7.0 service pack 2 included updated trusted CA certificates that work with the Basic Constraints feature. The certificates (democert.pem, democert1024.pem ca.pem, ca1024.pem, trusted.crt, demo.crt) are provided as files. If you are using a previous service pack of WebLogic Server 7.0 that contains expired trusted CA certificates, or demo certificates that do not work with the Basic Constraints feature, please see http://dev2dev.bea.com/products/wlserver81/wls_demo_cas.jsp for more information.
CR189285	When using the iPlanet LDAP Authentication provider, WebLogic Server maintains a pool of connections to the iPlanet LDAP server. These connections are actually connections to Big IP. Big IP is configured to close idle connection at regular intervals. In some cases, WebLogic Server would issue a query or command to the iPlanet LDAP server on a connection that Big IP closed. However, WebLogic Server did not produce an exception that detailed the problem. Now, if there are problems with connections to the iPlanet LDAP server, the LDAP server throws an exception that is propagated to the web tier. As a result, WebLogic Server now throws an exception message with the proper information.

Change Request Number	Description
CR205373	If a connection in the LDAP connection pool was invalid, an authenticate for the user was failing and a LoginException was being returned.
	Now, after a failed login with an existing LDAP connection from the connection pool, WebLogic Server creates a new connection in the pool and uses that connection. As a result, user requests no longer fail due to the LDAP connection timing out.
CR210229	Calling the `ServletAuthentication.weak()` method no longer results in an EmptyStack exception.
CR194073	Now, WebLogic Server processes the extendedkeyusage extension only when it is marked critical. As a result, WebLogic Server no longer throws a BAD_CERTIFICATE exception on the server side during two-way SSL communication.
CR045135	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-65.00.jsp.
CR125592 CR196597	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-66.00.jsp.
CR127930 CR107359	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp.
CR128940	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp.
CR157157	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-71.00.jsp.
CR171885	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_59.00.jsp.
CR172187	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-69.00.jsp.
CR175310	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_64.00.jsp.
CR175966	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-72.00.jsp.
CR128940	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp.

Servlets

Change Request Number	Description
CR120545	While configuring a custom xml parser in weblogic-applicaiton.xml, WebLogic Server no longer throws a Null Pointer Exception.
CR121297	WebLogic Server now forwards the error pages with the proper user.
CR112793	If the client dropped the connection, WebLogic Server was not processing the request any further.
	A system property, ProxyForConnectionResets, was added to WebLogic Server. By default it is set to false. When ProxyForConnectionResets is set to true, HttpClusterServlet will continue processing the request even if the client has broken the connection at some time during the processing.
	As a result, WebLogic Server processes requests even if the client drops the connection.
CR106348, CR125989	A failure in the listener left the Web Application in an unstable (deployed) state instead of undeploying the Web Application.
	When the listener fails (deployment is not completely successful), the Web Application deployment is rolled back.
CR134351	The flag, weblogic.http.client.defaultReadTimeout, has been introduced to set the default readtimeout on WebLogic Server's HttpURLConnection.
	When this flag is set, every new HttpURLConnection created on the JVM will have this value set as default. The value can be overridden by setTimeout or setReadTimeout calls on the specific connection. When this flag is not used, the default value is -1. The value should be provided in milliSeconds.
CR132447	If there is a cookie in a header that contains the SESSIONID name as a substring, the SESSIONID was unexpectedly being changed.
	This problem was resolved by correcting the logic in parsing SESSIONID from the cookie header.
	Due to this new change, no new SESSIONID is created when there is an existing SESSIONID or an existing session.
CR143448	The `java.lang.IllegalStateException: HttpSession is invalid` exception occurs in the servlet container's internal call. If other threads using the same session ID invalidate the session object during processing of ServletRequestImpl.syncSession(), an IllegalStateException may occur while calling SessionData.putValue() or SessionData.isNew().
	Ignore the IllegalStateException if the session has been invalidated by other threads.

Change Request Number	Description
CR128519, CR132321, CR130021	A ClassCastException occurred when using HttpProxyServlet with a Wrapped Response from a Servlet filter. A code fix ensures getting the original response in case of a Response Wrapper.
CR175651	A NullPointerException was being thrown when calling the `RequestProcessor.getServletContext()` method in a Struts-based application. To fix this problem, a system property, `weblogic.http.requestCompletionTimeoutSecs`, was added to WebLogic Server in the startup script file. The value given to this flag indicates the number of seconds for the container to wait before finishing all of the inflight work. The default value is 0 seconds, so the container does not wait if the flag is not present. The servlet container waits for the number of seconds specified in the value of this flag before it deploys or undeploys any web application.
CR184726	Clients that post a request that gets forwarded no longer lose the parameters when `weblogic.httpd.inputCharset` is set.
CR176941	The `GenericProxyServlet.readStatus` fails intermittently. This problem occurs if `GenericProxyServlet` reuses the connection that had already been closed by a backend server. This was resolved with a code fix to retry the same request when encountering the half-open socket exception and filtering out the connection header.
CR172672	When HttpClusterServlet tried to reuse a recycled connection that WebLogic Server had already closed, it would get a SocketException and mark the WebLogic Server as BAD. Now, if HttpClusterServlet gets a SocketException when using a recycled connection, it will make another attempt to connect to the same WebLogic Server with a new connection.
CR195698	Replacing the old WAR modules with new ones in the external_stage area was causing a 404 error. This problem was resolved by staging the actual application in the temp folder.
CR200450	Locale variants passed by the browser were causing unexpected results in validation rules in Weblogic Server. The locale header parsing logic was not considering the variant part of it. Now, WebLogic Server considers the locale variant while parsing the language header. The language header can be in the following format: `lang-country-variant;q=weight`

Change Request Number	Description
CR207481	If the incoming JSP request URI was transformed in the filter, the web container was generating duplicate JSPStubs for the same transformed URI, and therefore leaking memory. A code change fixed the problem such that the web container no longer creates duplicate JSPStubs.
CR188237	The issue with using custom output streams in the HttpServletReponse object when using HttpClusterServlet has been fixed. Now, clients can use custom OuputStreams in the HttpServletReponse object.
CR172214	WebLogic Server was undeploying a Web Application from the default HttpServer even if it was not deployed on the default HttpServer. Now, WebLogic Server undeploys the Web Application from the default HttpServer only if it is deployed on that HttpServer.
CR209387	Now, HTTPURLConnection makes an HTTP call only once (instead of twice) when making the call from a Java Client if the setTimeout() method is used in the Java Client.
CR211410	ChunkUtils.is2chunk is no longer stuck in an infinite loop. WebLogic Server performance has been improved as a result.
CR188953	JSP pages no longer hang when multiple users access a page cached using the weblogic.cache.filter.CacheFilter.
CR129064	Now, when an HTTP request is sent through the iPlanet plug-in, WebLogic Server is no longer incorrectly setting the dynamic server list to WebLogic Cluster 2.

Simple Network Management Protocol (SNMP)

Change Request Number	Description
CR109689, CR103678	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.

Web Services

Change Request Number	Description
CR127391	The SOAP HTTP Binding states that when an Exception occurs within a WebService, the server should throw a HTTP 500 "Internal Server Error", with a SOAP Fault response representing the exception.
	Many WebService clients use URLConnection in order to make the HttpConnection to the WebService. Weblogic Server overrode the java.net.HttpURLConnection with a different version. WebLogic's version threw an exception from the getInputStream method if the status code was >= 400.
	Since the exception was thrown there was no way for the WebService client to retrieve the inputstream, and thus it violated the Http SOAP Binding Specification.
	WebLogic Server now retrieves the inputstream upon a 500 error.
CR181695	Autotype was not handling inheritance correctly. The derived type did not extend the base type in the types.xml file.
	The inheritance for autotype has been added so that autotype is now handling inheritance correctly.
CR182069	Setting typeMappingFile with autotype did not work as expected.
	The problem has been fixed. Autotype now picks up the existing types from typeMappingFile.
CR175471 CR175536	Accessing an external web service via an outbound proxy server (iPlanet Web Proxy Server) returned a SOAP fault. This occurred because using http and https to invoke a remote web service through proxy server with proxy authentication turned on was not allowed.
	A code change was made to allow http and https tunneling through a proxy server for a web service client.
CR177611	There was a namespace problem in web-service.xml when servicegen used the type mapping file generated by autotype.
	servicegen now recognizes namespaces in the type mapping file generated by autotype.

Change Request Number	Description
CR183555	JAX-RPC clients did not maintain session stickiness using the SESSION_MAINTAIN_PROPERTY.
	WebLogic Server now supports session stickiness if SESSION_MAINTAIN_PROPERTY is set as follows:
	On the Call object:
	call.setProperty(Call.SESSION_MAINTAIN_PROPERTY, "true"); or call.setProperty(Call.SESSION_MAINTAIN_PROPERTY, new Boolean("true"));
	On the Stub object:
	(Stub)port)._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, "true"); or (Stub)port)._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, new Boolean("true"));
CR186319	When the number of contiguous white spaces in an xml file exceeded 32, WebLogic Server required a long time to parse the file.
	WebLogic Server now has a larger buffer size and parses xml files more efficiently.
CR197698	As a result of a client-side patch, webservice HTTPS client no longer fails with a java.net.UnknownHostException: null.
CR185173	When calling into a WebLogic Server 7.0.2 web service from a WebLogic Server 6.1.4 dynamic client, running the `webserviceclient+ssl.jar` file through the VersionMaker utility, and then using the modified jar file in the WebLogic Server 6.1.x environment, WebLogic Server no longer throws a `java.lang.NoClassDefFoundError`.

WLEC

Change Request Number	Description
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

Change Request Number	Description
CR136608	XA.End needs to be called before sending response to Tuxedo in WTC. WTC was sending a response to Tuxedo for TP/TPA calls before calling XA.End. This resulted in a race condition that could cause an XA.Start on the original thread to begin prior to XA.End being called.
	The solution now cleans all success and failure response paths and ensures that XA.end is called before sending any response back to Tuxedo.
CR172581	The following WTC FML methods in WebLogic Server 7.0 offer degraded performance as compared with the same functions executed under WLS 8.1. These functions include:
	• TypedFML32._tmpostrecv()
	• Fget()
	• Foccurs()
	• Fadd()
	These methods have been backported from WebLogic Server 8.1 to WebLogic Server 7.0 sp6.
CR175113	Thread starvation was causing a decrease in throughput. For an incoming request, WTC spawns a delegate thread that creates an object of type InboundEJBRequest and spawns another thread to execute that object. The threads are executed from the 'default' execute queue. If the EJB application that was executed by the InboundEJBRequest was slow compared to the rate at which input requests were arriving (from the remote Tuxedo domain), thread starvation could occur resulting in a backlog in 'default' execute request first and then a loss of input requests.
	The slow executing EJB application can now be executed in a dedicated thread pool. WTCService recognizes an execute queue named 'weblogic.wtc.applicationQueue' as a dedicated queue for executing WTC applications. If it is not preconfigured and if a system property 'weblogic.wtc.applicationQueueSize' is found to be set, this queue is created (during WTC boot period) explicitly.

Change Request Number	Description
CR177212	A NullPointerException was thrown for WTC request.
	WTC now checks the return value of `Utilities.xdr_decode_string` before instantiating the StringBuffer.
CR179410	WTC did not clean up the old connection after reconnect. Tuxedo /Domain and WTC are connected with ON_STARTUP connection policy. If the machine on which the Tuxedo domain was running was down, when it is restored, WTC did not clean up the old connection.
	Now, when the same remote domain receives a new connection, the old connection is dropped.

XML

Change Request Number	Description
CR120545	While configuring a custom xml parser in weblogic-applicaiton.xml, WebLogic Server no longer throws a NullPointerException.
CR135726	When WebLogic Server is unable to handle a given schema construct, it is mapped to SOAPElement.
	When this occurred, autotype gave no indication that it had encountered an un-supported schema construct and only examination of the generated classes showed that something had gone wrong.

WebLogic Server 7.0 Service Pack 5 Solutions

The following sections describe problems resolved for the release of WebLogic Server 7.0 Service Pack 5 (SP5). The following list of resolved problems is updated periodically.

-

-

-

-

-

Administration Console

Change Request Number	Description
CR091775	The Dispatch Policy can now be configured from the Administration Console. To configure this attribute, select 'Edit Webapp Descriptor' and the in the new window, select Webapp Ext. The Dispatch Policy attribute appears on the right.
CR126506	Users could not configure a console extension to replace the policy definition pages. Code was added to enable the `CreateResourcesAction` to look for an extension before forwarding.
CR132700, CR135127, CR174568	The `getParameter()` method in `GraphApplet` is used to get the value of the min/max heap size. WebLogic Server was using `Integer.parseInt` to convert the string values, and this parsing failed for numbers over 2G, resulting in incorrect values being displayed in the `applet-weblogic.management.console.applets.GraphApplet` on the Monitoring -> Performance tab. WebLogic Server now uses `Long.parseLong` to get the correct value of the heap even if it is over 2G.
CR100745	The deploy tab of a Web application showed that the application was deployed to the virtual host, which was true, and to the target of the virtual host, which was not true. A code change resolved the problem.

Classloaders

Change Request Number	Description
CR128510	The `readClassDescriptor()` of `MsgAbbrevInputStream` tried to resolve the class, leading to `ClassNotFoundException` for unknown classes. Java serialization skipped this `ClassNotFoundException` if corresponding data was not being read. The `MsgAbbreveInputStream` `readClassDescriptor()` no longer tries to resolve the class, and `MsgAbbrevInputStream` now implements `resolveClass()`.
CR124348	Client programs could not use `java.lang.reflect.Proxy` 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 `ClassNotFoundException`. The `resolveProxyClass()` method was implemented to load interfaces from the application-specific classloader as well as the system classloader.
CR111924	WebLogic Server threw a `ClassNotFoundException` 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. The problem was solved with a code fix.

Clusters

Change Request Number	Description

CR107471, CR128979, CR136731	When using HTTP tunneling with a cluster, the client got this message:
	
	`java weblogic.Admin -url http://colma:17683 -username system -password password PING 10`
	
	`<May 29, 2003 10:14:18 AM PDT> <Error> <RJVM> <000515> <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.receiveAndDispatch(HTTPClientJVMConnection.java:422)`
	
	`at weblogic.rjvm.http.HTTPClientJVMConnection.execute(HTTPClientJVMConnection.java:305)`
	
	`at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213 at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)> Failed to connect to <urldeleted> due to: <urldeleted>:Bootstrap to <urldeleted> 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 plug-in.

CR112874	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 were killed in that order, the client gave the following message: `<Jul 22, 2003 1:32:56 AM IST> <Warning> <Kernel> <No replica in list has the expected weight. Reverting to round-robin>` When the Managed Servers were restarted, the load balancing algorithm switched to round-robin. Analysis revealed that the replica list was updated when a Managed Server went down, but due to a race condition the max weight in RichReplicaList was not reset properly. A code change to recompute max weight whenever the replica list size changes solved the problem.

Change Request Number	Description
CR111029	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.
	The timeout value IdlePeriodsUntilTimeout is now tunable. It is set on the <Server IdlePeriodsUntilTimeout="3"> 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.
	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.
CR125966	A server instance served a replication request while it was being shut down, resulting in a `ClassCastException` during `syncSession`, because the Web container was trying to typecast the `ReplicatedSessionContext` to `MemorySessionContext`.
	Following a code change, a server instance can no longer serve a replication request when it is being shut down.
CR129234	`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. 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 this by passing the `sessionId` instead of ROID.
CR127849	On non-homogeneous unbinds and rebinds on cluster nodes, the server sometimes sent an uninitialized Remote Reference back to the client, resulting in an `AssertionError`.
	The problem was resolved by a code change that ensures that the primary representative is set to null on unbind.

Change Request Number	Description
CR112326	A memory leak occurred with `weblogic.cluster.BasicServiceOffer` during JNDI rebinds of replicable objects. The problem was resolved with a code correction.
CR116954	When `HttpClusterServet` sent a dummy request "/WLDummyInitJVMIDs" to get a cluster server list in the response header when there was no default Web application deployed, WebLogic Server logged a debug message like the following: `####<Sep 6, 2002 4:56:43 PM EDT> <Debug> <HTTP> <foo> <ms1foo> <ExecuteThread: '13' for queue: 'default'> <kernel identity> <> <101147> <HttpServer(887891,null default ctx,ms1foo) found no context for "/WLDummyInitJVMIDs". This request does not match the context path for any installed web applications and there is no default web application configured.>` A code change has resolved the problem.
CR121113	`HttpClusterServlet` did not print thread name in error messages, making it difficult to diagnose the problem when an error occurred. A code change was made to `GenericProxyServlet.trace()` to include the thread name in logged error messages to aid in troubleshooting.
CR127643, CR129319	When a dynamic proxy that implemented interfaces declared inside a Web application was put into the `HttpSession` and the session was replicable, WebLogic Server was not able to load the interface classes on the secondary server. To correctly resolve the dynamic proxy, the secondary server needs the name of the application where the interface resides. An `annotateProxyClass` was implemented in `MsgAbbrevOutputStream` to write the `applicationName` in the stream. On the receiving side, `resolveProxyClass` uses this application name to load the interface classes from the application. As a result of this change, dynamic proxies (implementing interfaces stored in the application archive) can be put into the `HttpSession` and be correctly replicated. There should be no side effects.

Connectors

Change Request Number	Description
CR131745	For Resource Adapters that specify transaction support of `LocalTransaction`, the transaction state for a connection was not being reset before being released back to the pool of available connections. If another thread picked up the thread immediately and tried to use it, it could result in the connection not being enlisted and shared properly. Following a code change, the transaction state is now immediately reset when the transaction is complete, which will occur before the connection is returned to the available pool.

Core Server

Change Request Number	Description
CR131692	In 7.0, the shutdown process did not discriminate between application level threads and threads performing work for the WebLogic Server infrastructure. Refusing requests for new connections resulted in unwanted WebLogic Server behavior. Some examples include: the Node Manager attempted to restart the server being shut down, session replication for current sessions failed, and the console reported that the server was in an unknown state. Code was added to assist the server in a graceful shutdown.
CR125245	The java.lang.LinkageError: duplicate class definition: error sometimes occurred when multiple threads attempted to load the same class in a Web application while PreferWebInfClasses was enabled. This problem occurred the first time the classes were loaded. A code change to synchronize the loadClass method of ChangeAwareClassloader resolved this problem.
CR103999	In some WebLogic Server installations the error "Unrecognized property: webservice.client.ssl.strictcertchecking" appeared. This error resulted from the inability of the installation to find the system property passed in its list of properties. The error message did not signify that the property was not taking effect, and as the property has been added to the properties list, the message should no longer appear.
CR103032	The Administration Console was updated to allow the user to edit the 'Allow Remove During Transaction' attribute for stateful session beans.

Change Request Number	Description
CR100705	There was an Administration Server dependency in the `SecurityConfigurationMBean` causing a failure in `SecurityConfigurationMBean.findDefaultRealm()` if the Administration Server was down.
	This was corrected by providing a mechanism to invoke operations on a local MBean server when the Administration Server is unavailable.
CR100373, CR130592	The logic to determine the type of J2EE-module did not use the local or staged copy of the application for Managed Servers. This problem surfaced in the MSI mode as a result of the server's inability to determine the J2EE-module type.
	`stagingLocation` is used to determine the type of J2EE-module if the original path does not exist in the `config.xml` file of the Managed Server.
CR122939	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.
	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.
CR130352	The WLServer ant task, backported from WebLogic Server 8.1 to 7.0 in Service Pack 4, sometimes created a `config.xml` file without the specified properties.
	Analysis revealed that WLServer never explicitly called `saveDomain`, which writes MBean changes to `config.xml`. Instead, it relied on the trigger calling `saveDomain`. The problem was that WLServer started and shut down a server so quickly that sometimes the trigger did not happen in time.
	This problem has been resolved by putting a `saveDomain` call into the WLServer ant task to force the `config.xml` to be written out before the server is shut down.

Change Request Number	Description
CR129002	WebLogic Server allows the configuration of JMS Distributed Destinations from the Administration Console even if the user does not configure any members for the distributed destinations. The server failed to reboot, however, when this configuration was invalid. Following a relaxation of legal checks in `JMSLegalHelper`, if a distributed destination is configured but members are not assigned, the server is no longer made unbootable.
CR128648	The `weblogic.deployer` WLConfig task was swallowing all `IllegalArgumentExceptions`, which caused build failures related to the encrypted system password functionality. The problem was resolved with a code fix.
CR128445	`weblogic.Admin` set the username and password for the `ServerStartMbean` each time it was used in a server instance. If a user attempted this it would fail without permissions to WRITE attributes. The problem was resolved with a code change that caused WRITE attributes not to be set on `ServerStart`.
CR126280	Rebooting the Administration Server caused a domain administration port exception. Code was added to the RMI boot service to check the MSI mode. This eliminated the exception.
CR125309	An incorrect return code status was returned from `weblogic.Admin` SHUTDOWN or FORCESHUTDOWN. Code was added to correct the return status from `weblogic.Admin` SHUTDOWN or FORCESHUTDOWN. **Note:** Applications relying on the incorrect status code may need to be re-written.
CR125285	`FileDistributionServlet` should return after calling the `sendError()` method. Code was added to return from all methods in `FileDistributionServlet` after `sendError()`, if that is not the last thing it does in a method.
CR125018	Certain common proxy operations were not happening on the local server where they should have been executed. Code was added to ensure that `userLockoutManager` calls are local by implementing a map of local calls. `UserLockoutManager` now returns the information for the local server, rather than for the Administration Server.

Change Request Number	Description
CR093109	Killing an entire WebLogic Server domain killed the Administration Server, instead of killing only the Managed Servers. This problem was solved with a code fix.
CR094219	Server and domain logging did not work as expected, in the following ways: The domain log file failed to rotate despite being configured to rotate by time. The "Limit Log Files" option did not work properly with `SimpleDateNamed` log files. A code change has corrected these issues, enforcing the log `fileCount` restrictions when using `SimpleDateFormat` to name log files.
CR094410	When DebugHttp was activated in `ServerDebugMBean`, WebLogic Server reported `SocketResetExceptions` as errors, rather than as simple debug messages. To address this problem, `logMuxableSocketResetException()` was added to the message catalog for reporting this debug message.
CR098578	`Bootstrap` servlet was returning SUCCESS status for the Administration Server when it was not actually fully up or was not in the RUNNING state. Following a code change, the `Bootstrap` servlet now attempts to get the `ServerRuntime` MBean of the Administration Server and checks its status to make sure the server is in RUNNING mode before returning a successful status. Managed Servers attempting to boot and contact the Administration Server will now return "Booted in MSI mode" if the Administration Server is not in the RUNNING state when the Managed Server is booted.
CR098607, CR136879	In a situation in which Application A is using Application B, while looking up Application B, the J2EE container of Application A creates a `DependencyClassLoader` by attaching Application B's `ClassFinder` to it. On redeployment of Application B, Application B encounters a new `ClassFinder` and its old `ClassFinder` is no longer valid. On a client request, the server attempts to use the old `DependencyClassLoader` and throws an exception. The behavior that caused the exception has been changed: WebLogic Server no longer uses `DependencyClassLoader` for loading impl class from `ReplicaAwareRemoteObject.getPrimaryRepresentative()`.
CR104269	If you configured a new customer authenticator in the Administration Console, setting the control flag and the default authenticator's control flag both to SUFFICIENT, then restarted the server, then used the `set attribute` command, and then restarted the server again, you received a security error message and were not allowed to start the server. The problem has been resolved with a code fix.

Change Request Number	Description
CR105338	WebLogic Server stopped logging, on both Administration Server and Managed Servers, when the maximum number of files was reached.

After the following error messages were printed out to the domain log file, the server logging service shut down.

```
####<Apr 21, 2003 3:17:39 PM GMT+09:00> <Alert> <Log
Management> <KESATO01> <myserver> <ExecuteThread: '10' for
queue: 'weblogic.kernel.Default'> <<anonymous>> <>
<BEA-170017> <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.> ####<Apr 21,
2003 3:17:40 PM GMT+09:00> <Critical> <Logging> <KESATO01>
<myserver> <ExecuteThread: '10' for queue:
'weblogic.kernel.Default'> <<anonymous>> <> <000000>
<Handler: 'C:\bea81ga\mydomain\myserver\myserver.log'
raised several exceptions. Shutting it down> ####<Apr 21,
2003 3:17:40 PM GMT+09:00> <Error> <Logging> <KESATO01>
<myserver> <ExecuteThread: '10' for queue:
'weblogic.kernel.Default'> <<anonymous>> <> <000000>
<Handler: 'C:\bea81ga\mydomain\myserver\myserver.log'
raised exception when opening. Exception
weblogic.logging.LogRotationException null> ####<Apr 21,
2003 3:17:40 PM GMT+09:00> <Error> <Logging> <KESATO01>
<myserver> <ExecuteThread: '10' for queue:
'weblogic.kernel.Default'> <<anonymous>> <> <000000>
<Handler: 'C:\bea81ga\mydomain\myserver\myserver.log'
raised exception when opening. Exception
weblogic.logging.LogRotationException null>
```

If the problem occurred, all log messages were not printed out to the server log file.

Analysis revealed that WebLogic Server closed log files during rotation and, if log rotation failed, the log files would remain closed.

Log rotation failed when WebLogic Server generated the wrong index for the new log file. Because it assumed that the file that had the latest time stamp was the latest index, it did not check whether a file with the index name 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.

In addition, for time-based log rotation, on multi-CPU machines, multiple rotations happened within the same millisecond.

The problems were resolved with code fixes.

Change Request Number	Description
CR105516	In previous WebLogic Server 7.0 service packs, stateful session EJB failover did not work when multiple failovers were required.
	The following call sequence with the same browser window led to a `java.rmi.ConnectException` when only one node of the cluster survived:
	1. All three cluster nodes are running.
	2. Making a call to node1, create EJB and store remote in http session (HTTP session replication is enabled).
	3. Kill node1.
	4. Make a call to the secondary node2, the EJB remote is retrieved from the replicated HTTP session and the call to the EJB is successful. After this EJB call again the remote is stored in the HTTP session.
	5. Kill node2 .
	6. Make a call to node3 and get the EJB remote from HTTP session.
	WebLogic Server tried to look up the EJB on node2 and did not try to use node3. The following exception was thrown on node3:

	`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(B`
	`asicRemoteRef.java:97) at`
	`weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(Replica`
	`AwareRemoteRef.java:255)`

	The problem was solved with a code fix.

Change Request Number	Description
CR106957	In earlier service packs of WebLogic Server 7.0, 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 Windows, if Native IO (Performance Pack) was disabled, or if the `libmuxer.so` library was removed from the classpath.
	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.
CR108727	In previous service packs of WebLogic Server 7.0, 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 the `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.
CR109307	Previously you could not bring up an SP2 Managed Server in an SP3 domain. A code change has brought the different service packs into compatibility and into accord with the WebLogic Server compatibility statement: "Servers within an Administrative domain can be at different Service Pack levels as long as the Administration Server is at the same Service Pack Level or higher than its Managed Servers."
CR109391, CR123398, CR174565	In previous service packs, WebLogic Server attempted to deserialize even nonserializable objects if they were put in the `ServletContext`, resulting in `NotSerializableException`.
	Non-serializable objects are no longer deserialized in `ServletContext`.
CR109688, CR135235	Under certain conditions a simple Java client looking up `InitialContext` would receive a bogus "No version information" error.
	The problem was that line breaks in the header buffer were marked by "\n" instead of by the `line.separator` property. A code change corrected the problem.

Change Request Number	Description
CR110233, CR121682	Server instances running in production mode inappropriately checked the `\applications` folder and created entries in temp files recording such checks. In development mode an application automatically gets undeployed and redeployed in the `\applications` folder at server start time.
	A code fix has resolved the problem.
CR111239	Lost JMS connections were not being cleaned up. The clients (topic subscribers) were running on a different machine than the JMS server. When the customer's network connection between a client and the server was lost, after closure and reestablishment of the subscriber, the session, and the connection the Administration Console showed that the first subscriber was not removed—the console showed two subscribers. When the `<MessagesMaximum>` value for the Connection Factory is reached, the 'messages pending' value increases with each new message published.
	Analysis revealed that due to the network outage the client's `peerGone` message did not reach reached the server instance. After declaring peergone the client tried to reconnect to the server instance, which replaced the old connection with the new one without declaring a `peerGone` on the old rjvm, thus causing the JMS objects to remain in memory.
	The problem was corrected with a code change that causes the server, when detecting a duplicate connection from the same client for the same protocol, to declare `peerGone` on the old rjvm and construct a new rjvm and a new `ConnectionManager` for the new connection.
CR116707	WebLogic Server provided a `ReplicaAwareRemoteRef` warning log when you used certain Factory/Trader patterns, displaying the following on the client side:
	`[ReplicaAwareRemoteRef] : WARNING: client-side RA stub didn't find environment on thread`
	Following a code change, the inappropriate log is no longer printed to the client.
CR116712	In previous service packs, the `WLECConnectionRuntime` MBeans were not updated when an INVOKE was issued on `WLECConnectionPoolRuntime` using the Admin tool.
	Analysis revealed that when `resetConnectionPool()` was invoked on the `WLECConnectionPoolRuntime` MBean, the pool was not reset. Old connections were not removed.
	The problem was solved with a code change. Now, the MBeans that correspond to old connections are unregistered before the `resetConnectionPool()` invocation.

Change Request Number	Description
CR120492, CR122551	To improve HTTP tunneling performance, a code change has set the content length for tunneled responses to maintain keep-alive connections, and has also enabled caching HTTP URL connections.
CR121801	The wlconfig tool, introduced in WebLogic Server 8.1, is available in 7.0 starting with Service Pack 5.
CR134971	When `<<no stack trace available>>` was sent out as part of the exception message field (from the server side), the 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).
	WebLogic Server now parses the exception message field, traps this specific exception and ensures that it is appended only once.

Change Request Number	Description
CR122112	When calls were made to an EJB using the `weblogic.management.timer.Timer` class, the following exception was thrown upon lookup of the EJB's home interface:

```
java.lang.ClassCastException: Cannot narrow remote object
to examples.ejb20.timer.SourceDataHome at
weblogic.iiop.PortableRemoteObjectDelegateImpl.narrow(Por
tableRemoteObjectDelegateImpl.java:200) at
javax.rmi.PortableRemoteObject.narrow(PortableRemoteObjec
t.java:132) at
examples.ejb20.timer.webapp.TimerServlet.callEjb(TimerSer
vlet.java:61) at
examples.ejb20.timer.webapp.TimerServlet.access$000(Timer
Servlet.java:23) at
examples.ejb20.timer.webapp.TimerServlet$MyNotificationLi
stener.handleNotification(TimerServlet.java:81) at
weblogic.time.common.internal.TimerListener$1.run(TimerLi
stener.java:48) at
weblogic.security.service.SecurityServiceManager.runAs(Se
curityServiceManager.java:685) at
weblogic.time.common.internal.TimerListener.deliverNotifi
cation(TimerListener.java:44) at
weblogic.management.timer.Timer.deliverNotifications(Time
r.java:382) at
weblogic.time.common.internal.TimerNotification$1.run(Tim
erNotification.java:112) at
weblogic.security.service.SecurityServiceManager.runAs(Se
curityServiceManager.java:685) at
weblogic.time.common.internal.TimerNotification.execute(T
imerNotification.java:109) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:
234) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:210)
```

Analysis revealed that WebLogic Server did not store the `ContextClassLoader` of the callee inside the Timer class. Therefore, the notification runs in a classloader which is different from the classloader in which the notification was added. This resulted in a `ClassCastException`.

The problem was solved by a code change to save the `ContextClassLoader` of the callee when the notification gets added. While delivering the notification event, the saved classloader is set as the `contextClassLoader` for the delivering `ExecuteThread`.

Change Request Number	Description

CR122361	The Auditing enabled message did not record the USER identity:
	
	`<Aug 29, 2003 12:31:25 AM EDT> <Info> <Configuration Audit>`
	`<159909> <Configuration Auditing is enabled>`
	
	This was inconsistent with the disable message, which did record the USER:
	
	`<Aug 29, 2003 12:31:31 AM EDT> <Info> <Configuration Audit>`
	`<159910> <USER system, Configuration Auditing is disabled>`
	
	Following a code change, both audit messages record the USER identity.

CR122706	A clustered Web application deployed on a cluster that called an EJB on another cluster threw an `AssertionError`. Analysis revealed that generated code was not using the right method while calculating methods in the interfaces, which resulted in using wrong classloader from the generated stub. A code change resolved the problem.
CR122878	MBeans were being deleted by KernelID. As a result, no matter who initiated the delete operation, the Auditor recorded that the Kernel had deleted the MBean. Following a code change, the MBean is deleted by the current Authenticated Subject.
CR123571	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 and cause exceptions or hanging of 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.
CR129094, CR133591, CR135254, CR135255, CR124861, CR132275	Performance issues that involved TCP window shrinkage in the t3 protocol on the AIX platform have been resolved.
CR135488	Previously, any file in the security provider directory (by default `/lib/mbeantypes`) was treated as a security provider and loaded at server boot time without regard to its file extension. Beginning with WebLogic Server 7.0 Service Pack 5, only files that end in `.jar` or `.zip` are treated as security providers. Security providers that use other extensions will no longer load. If your domain uses alternate file extensions for security providers, change them for Service Pack 5.

Deployment

Change Request Number	Description
CR110897, CR125329, CR129494	If you deployed an application and then unprepared it and then restarted the server, the server would throw the `IllegalArgumentException`.
	Analysis revealed that the J2EE container did not take care of configured but not deployed applications during start up. They were identified as Deployed="false" in the <Application> stanza.
	The J2EE container code has been modified to take care of this edge condition.
CR127393	The following memory leaks were identified:
	1) A memory leak with EJB Objects. `ClientRuntimeDescriptors` of EO objects remained on the heap after an application was undeployed.
	2) A memory leak with EJBCache in J2EEApplicationContainer. EJBCache objects were not being cleaned up after an application was undeployed.
	3) A memory leak with runtime MBean in `WebAppServletContext`. Runtime MBeans specific to `WebAppServletContexts` are not being unregistered after undeploying an application.
	4) A memory leak with Timer in `KeepAliveCache`. `KeepAliveCache's` timer object held an application-specific classloader.
	5) A memory leak with ddMap in J2EEApplicationContainerFactory. Descriptor objects were not being unregistered after an application was undeployed.
	6 A memory leak due to Introspector cache. Application-specific classes were being held by the Introspector caches after the undeploying of an application that uses Struts.
	Code was added to perform the following functions:
	1) Unexporting EO objects now removes the `ClientRuntimeDescriptor` objects associated with the EO objects on undeploying application.
	2) Clear EJBCaches on undeploying application.
	3) Unregister runtime MBeans specific to `WebAppServletContext` on undeploying application.
	4) `KeepAliveCache` will initialize the timer before it comes to the context of an application.
	5) Clear deployment descriptors on undeploying an application.
	6) Flush Introspector caches on undeploying an application.

Change Request Number	Description
CR136176	Setting `LoadBeforeAppActivation="true"` resulted in the `StartupClass` being invoked twice.
	Code was added to make the server ignore cases which are not applicable to `PostDeploymentStartupClass`
CR102296	A boolean attribute, `LoadBeforeAppActivation`, was added in `StartupClassMBean` so that there are three distinct settings for the order in which startup classes are deployed:
	Default: After a server instance deploys JMS and JDBC services, EJBs, and applications.
	`LoadBeforeAppActivation=true`: After a server instance deploys JMS and JDBC services and before it deploys EJBs and applications.
	`LoadBeforeAppDeployment=true`: Before a server instance deploys JMS and JDBC services, EJBs, and applications.
CR109645	When an application was configured for the second time after deletion, components within the application did not have the correct targets. Changes to the admin MBean were not being propagated to the config MBeans.
	When updating the config MBeans, if the target is a cluster, WebLogic Server now gets the servers in the cluster and for each server builds the objectname for the config MBean based on the server name and then updates it.
CR111065	In certain instances, the delta list of deployment tasks was being created from the `SlaveDeployer` initialization.
	A code fix ensures that the `MasterDeployer` is always used for creating the deployment task deltas.
CR134122, CR134119	The `StatefulEJBObject.remove()` call was unexporting an EJB object even if there was no permission for that method.
	This problem has been resolved with a code change.
CR128932	The Administration Console reported that an application targeted to a cluster, one of whose Managed Servers was gracefully shut down, was not deployed.
	A code change has resolved the problem.

EJB

Change Request Number	Description
CR127334, CR133975	A memory leak occurred when `EOImpl` instances of the EJB were not garbage-collected when `remove()` was called on an already passivated and deleted EJB. A code change has resolved the issue.
CR132510	A collection valued `cmr-field` was accessed in a different transaction from the one in which it was created. Code was added to check if the current transaction is the same as the transaction in which the generated `oneToManySet` was created (`createTx`). If it is not, then set the `createTx` on the `oneToManySet` to the current transaction.
CR131848	A `ClassCastException` was thrown at runtime when "cache-between-transactions" is set to true for a BMP `bean.java.rmi.RemoteException: EJB`. A check for BMP and CMP20 beans was added to prevent the `ClassCastException`.
CR103038	When Allow Remove During Transaction was set to "False," and an application attempted to remove a stateful session bean during a transaction, an inaccurate error message appeared. `weblogic.ejb20.locks.LockTimedOutException` is not the exception called for in the EJB 2.0 specification, and it has been replaced with the exception required: `javax.ejb.RemoveException`
CR132853, CR128980, CR135722	When a Message Driven Bean uses the synchronous message polling scheme and the Sonic JMS server is used, the message-driven bean container's polling optimizations could result in a delay in the message receiving. To avoid this problem, do not use the optimized poller. As the Sonic message delivery scheme does not work well with this scheme, use a poller that continuously polls the Sonic JMS server. This new Message Driven Bean behavior is applicable to TIBCO and Sonic JMS providers only.
CR133602	WebLogic Server uses a special handling code for SonicMQ 4.* version, when XAconnections are used. This code was not required for Sonic 5.* version. For this reason, a connection leak occurred. Avoid the use of special code for Sonic 5.* version inside the WebLogic Server MDB container.

Change Request Number	Description
CR133774	The following part of the description for the `max-beans-in-cache` element in the `weblogic-application.xml` DTD was incorrect: If 0 is specified, then there is no limit for max-beans-in-cache. A `max-beans-in-cache` size of 0, would actually set the actual size of the cache to 0, thus causing a `CacheFullException`.
	The above note has been removed from the DTD. Also, a compliance check has been added to not allow a value of 0 to be set for the `max-beans-in-cache` in the `weblogic-application.xml`. If you need to set a large size (infinite) for your cache, you can set so by setting the value of the max-beans-in-cache to `java.lang.Long.MAX_VALUE` in the DDs. This makes it compliant with the behavior of the same element in the `weblogic-ejb-jar.xml`.
CR126413	A CMP bean threw a `ClassCastException` while persisting a blob/clob cmp-field to the database when using WebLogic JDriver (type 2).
	Code was added to handle casting of LOBs (Blob & Clob) for both Oracle type4 and WebLogic type2 drivers.
CR129185	WebLogic Server loaded optimistic-concurrency beans in a separate transaction, suspending the current transaction, starting a new one, loading the bean and resuming the old transaction for all databases except Oracle, to avoid acquiring locks during the SELECT. Since the default behavior in Sybase is for a a shared-lock to be acquired during the SELECT but released even before the statement has completed the statement, the loading behavior has been changed: When the concurrency-strategy is optimistic, WebLogic Server now suspends and resumes the transaction for databases other than Oracle only if the isolation-level is higher than Read-Uncommitted or Read-Committed.
CR127397	Entity beans use `ActivatableServerRef` to call activate to get the bean instance. After invoking the bean method, `ActivatableServerRef` calls deactivate to release the bean to the pool. The problem was that when a EJB was invoked from within the application, deactivate was not called and the bean was not released.
	`ActivatableServerRef` now explicitly calls `notifyRemoteCallBegin` and `notifyRemoteCallEnd`.
CR127361	`EJBReplacer` did not consider remote objects and hence threw a `NotSerializableException` while passivating a bean that contained a remote object as a member variable.
	Following a code change, `EJBReplacer` replaces the remote object as appropriate if the bean instance contains a remote object member variable.

Change Request Number	Description
CR079164	A less than useful deployment error message that the server displayed on an error setting up auto-pk generator table has been replaced with a more detailed one that gives the nested exception message with more clarity and details.
CR093520	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 recompiled 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. The problem was resolved by storing the timezone at compile time and using that timezone at deployment time to determine whether recompilation is necessary.
CR106136	In previous WebLogic Server 7.0 service packs, getter methods for an EJB 1.1 CMP bean did not call `isModified()` when `delay-updates-until-end-of-tx` was set to false A session EJB called an entity EJB's getter methods. Both EJBs had container-managed transactions with the transaction attribute set to Required. Each call to a getter method is followed by a call to `ejbStore()`, and `delay-updates-until-end-of-tx` was false. However, before calling `ejbStore` on the bean, the container did not call the `isModified` method until the transaction committed. `ejbStore` is now called from `postInvoke()`, depending on the result of the `isModified` method in the bean. This has resolved the problem.
CR108084	When a sender posted a JMS message, triggering a message-driven bean, if the MDB never obtained the data in the database it would drop the message. The problem was caused by a known Oracle XA limitation wherein an open resultset is invalidated by a suspended transaction. A code change resolved the problem.

Change Request Number	Description
CR110917	An `OutOfMemoryError` occurred when the WebLogic Server stateful session bean example (`examples.ejb20.basic.statefulSession`) ran under heavy load.
	The out-of-the-box configuration had been modified to set:
	`home-is-clusterable = true`
	`replication-type = InMemory`
	The application archive was deployed to a cluster of two Managed Servers. A Java client performed these steps:
	• Look up the home.
	• Create an instance of the EJB.
	• Call the buy and sell methods.
	• Invoke the remove method on the EJB.
	Loading the cluster with such requests resulted in the out-of-memory condition. When the client actions were stopped, the server instance did not recover—the console continued to report high heap usage.
	Analysis revealed the secondary EJBObjects were not being unexported, resulting in a memory leak.
	The problem was solved with a code change to unexport the secondary EJBObjects.
CR111551	In previous service packs of WebLogic Server 7.0, EJBC generated CMP code that caused a `java.lang.ClassCastException: oracle.sql.BLOB` error.
	Analysis revealed that the CMP RDBMS code generated for the query was incorrect. Prior to WebLogic Server 7.0 SP05, server-side JDBC went through the RMI drive, to the pool driver, then on to the DBMS driver.
	The problem was solved by correcting the EJBC to generate code that reflects the SP05 and later behavior. Starting in WebLogic Server 7.0 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 `oracle.sql.BLOB` instead of `weblogic.jdbc.common.OracleBlob`.

Change Request Number	Description
CR111670	After an unsuccessful attempt to create a `connectionPool`—for instance, using an invalid URL or driver class—it was not possible to create the pool successfully.
	This problem occurred because the `connectionPool` MBean was created before the `connectionPool` was created, regardless of whether the `connectionPool` was successfully created. Subsequent attempts to create a pool failed during the attempt to create a second MBean of the same type.
	The problem was solved by modifying the `JDBCService` to delete the associated MBean if an exception creating a `connectionPool`.
CR112074	When an update call followed by a remove call was performed on a container-managed relationship in a business method, `ejbStore` was invoked in the middle of the transaction.
	The extra `ejbStore` occurred because the `flushModifiedBeans()` method was called inside the `BaseEntityManager.caseDeleteRemove()` method.
	The problem was solved with a code change to ensure that the `flushModifiedBeans()` method is called on for cascade deletes.
CR112558	Failover did not work for a stateful session bean in a cluster with the following exception.
| | ```
Start server side stack trace:
java.rmi.NoSuchObjectException: Unable to locate EJBHome:
'de.roland24.common.system.session.BackendSessionHome' on
server:
't3://192.168.201.52:0
at weblogic.ejb20.internal.HomeHandleImpl.getEJBHome(Home
HandleImpl.java:78)
at
weblogic.ejb20.internal.HandleImpl.getEJBObject(HandleImpl.
java:188)
at
weblogic.servlet.internal.session.SessionData.getAttribute(
SessionData.java:426)
``` |
| | A code change solved the problem. |

Change Request Number	Description
CR112265	The following exception is no longer received when container-managed persistence EJBs are built using EJBGEN tags: `<Jul 17, 2003 2:49:20 PM MDT> <Info> <EJB> <010097> <Exception during the invocation of EJB "Customer(Application: classproject, EJBComponent: xpersistence_ejb. jar)" with primary key "[CustomerPK guidStr:4E9A50DD-8E1B-F2B5-24E7-860ADE3BE707]": java.lang.NullPointerException at com.zbc.las.commercial.persistence.test.CustomerPK.hashCode(CustomerPK.java:24) at com.zbc.las.commercial.persistence.test.CustomerPK.equals(CustomerPK.java:32) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__ WebLogic_CMP_RDBMS_customerOrders_Set.add(CustomerBean_b3dzi6_ _WebLogic_CMP_RDBMS_customerOrders_Set.java:289) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__ WebLogic_CMP_RDBMS_customerOrders_Set.addAll(CustomerBean_ b3dzi6__WebLogic_CMP_RDBMS_customerOrders_Set.java:328) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__ WebLogic_CMP_RDBMS.setCustomerOrders(CustomerBean_b3dzi6__ WebLogic_CMP_RDBMS.java:672) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6_ ELOImpl.setCustomerOrders(CustomerBean_b3dzi6_ELOImpl.java: 268) at java.lang.reflect.Method.invoke(Native Method) at com.zbc.las.framework.core.persistence.EjbHelper.setRelations(EjbHelper.java:594) at com.zbc.las.framework.core.persistence.EjbHelper.setRelations(EjbHelper.java:351) at com.zbc.las.commercial.persistence.test.CustomerBean. setOrders(CustomerBean.java:302) at ....`
CR112703	The `JMSConnectionPoller` did not know the identity of the user that it required to get the remote username and password. This remote username and password is required for establishing a JMS connection with the remote JMS Server (MQ, WebLogic, and so on). This fix resolves the problem, but customers need to configure a security identity for an MDB. For instructions on how to do this, see Configuring a Security Identity for Message-Driven Beans in *Programming WebLogic Enterprise JavaBeans*.

Change Request Number	Description
CR112838, CR112225	For BLOBs, in the generated code, WebLogic Server calls `ObjectOutputStream.writeObject` and `ObjectInputStream.readObject` to serialize or deserialize the object before writing or reading it to the database. These calls add extra header information. The `writeObject` 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. These calls do not cause a problem when customers are using only WebLogic Server to set and get BLOBs, because WebLogic Server uses `readObject` to convert the bytes into the appropriate object, which needs the extra header information. However, if the BLOB has been inserted directly into the database by some other vendor or programmer using: `OutputStream os = ((weblogic.jdbc.common.OracleBlob) lob).getBinaryOutputStream(); os.write(this.tiffImage); // byte[] tiffImage` then problems may occur because WebLogic Server uses `readObject` 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. `<serialize-byte-array-to-oracle-blob>` has been added to control the persistence behavior. This element is used to specify whether a cmp-field of type byte[] mapped to an OracleBlob should be serialized. The tag has been added to a new compatibility stanza in the `weblogic-cmp-rdbms` descriptor. Note that, in versions prior to SP2, the default behavior was to serialize a cmp-field of type byte[] mapped to an `OracleBlob`. Now, the byte[] is written directly to the `OutputStream` obtained from the BLOB. To revert to the old behavior, set the value of this tag to true.
CR113161, CR12220, CR110917	A memory leak no longer occurs for a clusterable stateful EJB in the core of the server.
CR116696, CR121886	The `setBytes()` method on the preparedStatement had issues when the byte[] data was greater than 4K. This problem was resolved by replacing the `setBinaryStream()` method with the `setBytes()` method. As a result, the cmp11 generated code has changed. Customers wanting to take advantage of this fix must rerun ejbc on their EJBs to generate the new code. Otherwise, this change will not have any impact.
CR120450	A `java.lang.IndexOutOfBoundsException` no longer occurs when a CMP-field and a CMR-field were mapped to the same column.

Change Request Number	Description
CR121378	In WebLogic Server 7.0 SP03, when using bean-managed persistence (BMP) EJBs, a ClassCastException occurred when trying to flushModified Beans using BMP beans.
	<pre><Aug 25, 2003 9:39:35 AM EDT> <Info> <EJB> <010040> <Exception
in ejbStore: java.lang.ClassCastException:
com.logica.mdch.morse.business.meterpoints.MeterPointXBean_
vm8ya9_Impl java.lang.ClassCastException:
com.logica.mdch.morse.business.meterpoints.MeterPointXBean_
vm8ya9_Impl at
weblogic.ejb20.manager.ExclusiveEntityManager.flushModified(
ExclusiveEntityM anager.java:693) at
weblogic.ejb20.internal.TxManager$TxListener.flushModified
Keys(TxManager.jav a:749) at
weblogic.ejb20.internal.TxManager.flushModifiedBeans(
TxManager.java:329) at
weblogic.ejb20.manager.BaseEntityManager.flushModifiedBeans(
BaseEntityManage r.java:1644) at [...]</pre>
	Analysis revealed that the exception occurred when a BMP with exclusive concurrency that was involved in a transaction with other EJBs was synchronizing the modified beans with the database prior to before running any finder. ExclusiveEntityManger's flushModified code was not considering the BMP case, and was assuming container-managed persistence as the default.
	The problem was resolved to avoid the ClasscastException. As a result of the change, the BMP EJB's ejbStore method may be called more than once in a transaction, when a finder is invoked on any bean with in the TX.
CR122052, CR110440, CR121949, CR132495	For an entity cache timeout with a combined (application level) cache, if the idle-timeout-seconds for one of the beans is set to 0, other beans with idle-timeout-seconds > 0 are also not removed from the cache after timeout.
	A code change involving the registration of idle-timeout-seconds has resolved the problem.
CR123213	Stateless session beans were not returned to the free pool if they were invoked from the BMP ejbRemove. As a result, the pool reached maximum size and timed out while waiting to get an instance.
	A code change resolved the problem.

JCOM

Change Request Number	Description
CR095485	A COM object that closed during a load could result in sockets being left in a `CLOSE_WAIT` state. The sockets now close on `endOfStream`.

JDBC

Change Request Number	Description
CR120455	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.
CR127460	For the WebLogic Server jDriver for Oracle versions 817, 901 and 920 with rmi_jndi and rmi_driver tiers, multiple `ClassCastExceptions` were thrown in connection with the closing of the JDBCHelper class. A code change has resolved the problem.
CR126511	Oracle OCI NativeXA does not allow a connection to be used by threads other than the thread where it was created. The WebLogic Server connection pooling algorithm assumed that a connection can be used by any thread. If you used Oracle OCI NativeXA with the previous WebLogic Server connection pooling algorithm, you received an `XAException`. A code change backports the PinnedToThread connection pooling algorithm from WebLogic Server 8.1, allowing connections to be pinned to threads so that every thread can have its own connection for every connection pool.

Change Request Number	Description
CR108826	Current Multipool does not provide an option to define time based failover. We would like the exact condition under which failover will be triggered to a secondary connection pool. It would go through all the connections in the pool before it decides that it doesn't have any good connection.
	In order to avoid network glitches, we'd like have time based option like if all you checks fail for 10 minutes then failover to primary Database. We also need a failback mechanism through which we can failback to the primary once primary is in normal state. For example we have our secondary in remote datacenter, so response time will be more when in running in remote, we can live with when the primary database is down, but not when primary is backup
	So in summary, if my primary Database goes down, it needs to wait for xx time before my connection failover to secondary Database.
	Once my primary database is available then it should not go back to primary again until I say so. Either we can have an option which says what to do or run time check when these flag is turned on go back to primary or something like that.
CR130306, CR135909	In `jta.DataSource`, when doing `refreshAndEnlist`, WebLogic Server called `tx.enlist()`, but the connection was not returned to the pool if there was an exception in the `refreshAndEnlist` call.
	A code change catches the exception and releases the connection to the pool.
CR129379	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.
	This behavior has been changed so that the session need not hold cursors until the transaction ends.
CR127949, CR131743	`Statement.getResultSet()` sometimes generated an unnecessary new `ResultSet` wrapper.
	A code change has resolved the problem.
CR127720	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 `XAER_PROTO` or `XAER_RMERR` when an `xa_start()` 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.
	The problem was resolved by a code change in the recovery method that prevents special XA connections from being released to the pool twice.

Change Request Number	Description
CR126808	When two applications with an application scoped datasource per application were deployed on the same server and the datasources have the same name, an `InstanceAlreadyExistsException` was thrown.
	Code was added to configure the pool name to be application-specific when necessary.
CR133835	When statements or result sets that are not closed are garbage-collected, their `finalize()` method closes them if needed. A method that threw a useless exception if the connection had already been closed is no longer thrown.
CR099363	Under load testing, when the connection pool had refresh minutes set to 15 but the test connections on reserve and test connections on release set to false, the JDBC connection pool threw a `weblogic.common.ResourceException: No available connections in pool` exception. This occurred because when only a few of the connections in the pool are used, all the other connections in the pool were reserved for testing at the expiration of the configured refresh test minutes.
	A code change implemented a new algorithm that resolved the situation.
CR103603	When calling `DatabaseMetaData meta = dbCon.getMetaData();ResultSet rs = meta.getColumns(null,meta.getUserName(),"dual".toUpperCase()," %");`
	in a user transaction or inside a transactional EJB, if the `ResultSet` was not explicitly closed when the connection is closed, then each subsequent call leaked an Oracle database cursor. The cursors are held until the server shuts down, creating a risk that the database will run out of cursors.
	WebLogic Server now explicitly closes a `ResultSet` when it closes a connection.
CR108478	During a global transaction, when the client put a message into a JMS Queue and a database, the client looped repeatedly and then exited. If after this you restarted WebLogic Server following a crash (`kill -9` or `System.exit()`), when the client started a global transaction again, the following exception was thrown:
	`Unknown Exception during dataInsert java.sql.SQLException: XA error: XAER_PROTO`
	The addition of more robust error-handling code corrected the recovery behavior.

Change Request Number	Description
CR108826	In WebLogic Server 7.0SP5, the following enhancements were made to MultiPools: • Connection request routing enhancements. • Automatic failback on recovery of a failed connection pool. • Failover for busy connection pools within a MultiPools. • Failover and failback callbacks for MultiPools. See MultiPool Failover Enhancements in *Programming WebLogic JDBC* for more details.
CR110486	Remote clients can now clear the JDB connection prepare statement cache. Previously, objects of class `weblogic.jdbc.rmi.SerialConnection` did not implement `WLConnection`) so a remote client could not call the `clearPrepareStatementCache()` method. This limitation was corrected with a code change.

Change Request Number	Description

CR112295	A DataSource for a MultiPool could be deployed to an Administration Server; however, attempts to deploy it to a Managed Server failed, resulting in this `ResourceException`:
	
	`weblogic.common.ResourceException: DataSource(multiDataSource)`
	`can't be created with non-existent Pool (connection or multi)`
	`(multiPool) at`
	`weblogic.jdbc.common.internal.DataSourceManager.createDataSour`
	`ce(DataSourceManager.java:251) at`
	`weblogic.jdbc.common.internal.DataSourceManager.createAndStart`
	`DataSource(DataSourceManager.java:106) at`
	`weblogic.jdbc.common.internal.JDBCService.addDeployment(JDBCSe`
	`rvice.java:190) at`
	`weblogic.management.mbeans.custom.DeploymentTarget.addDeployme`
	`nt(DeploymentTarget.java:330) at`
	`weblogic.management.mbeans.custom.DeploymentTarget.addDeployme`
	`nts(DeploymentTarget.java:590) at`
	`weblogic.management.mbeans.custom.DeploymentTarget.updateServe`
	`rDeployments(DeploymentTarget.java:568) at`
	`weblogic.management.mbeans.custom.DeploymentTarget.updateDeplo`
	`yments(DeploymentTarget.java:240) at`
	`sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at`
	`sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess`
	`orImpl.java:39) at`
	`sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth`
	`odAccessorImpl.java:25) at`
	`java.lang.reflect.Method.invoke(Method.java:324) at ...`
	
	Analysis revealed that the exception occurred during the `DataSourceManager.validateConnectionPool()` method. The problem was resolved with a code change.

CR117729	With `weblogic.JTAJDBC` enabled, WebLogic Server prints out the driver vendor version as in the following:
	
	`<Aug 5, 2003 3:22:55 PM PDT> <Debug> <JDBC XA> <000000> <`
	`-tx:null- -pool:OracleThinXAPool- Vendor:Oracle 8.1.7 XA>`
	
	For Oracle, WebLogic Server printed the version as "Vendor:Oracle8.1.7 XA" regardless of the actual version.
	
	A code change has resolved the problem.

Change Request Number	Description

CR120531, CR126189	In previous service packs of WebLogic Server 7.0, resetting a connection pool using `weblogic.Admin RESET_POOL` (or an equivalent API) caused an exception similar to the following if a connection was already released:
	
	`<Aug 13, 2003 1:33:11 PM EST> <Error> <HTTP>`
	`<[WebAppServletContext(7096795,jdbc_webapp,/jdbc_webapp)]`
	`Servlet failed with Exception java.lang.Error: 1 Was already`
	`released:weblogic.jdbc.common.internal.Connection`
	
	After a garbage collection, the server would then display a connection leak
	
	warning:
	
	`<Aug 13, 2003 1:33:19 PM EST> <Warning> <JDBC> <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.<init>(Connection.java:55)`
	`        at`
	`[...]`
	
	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."

CR120971	The `jts.Connection.getVendorConnection()` method returned null when the vendor connection was obtained before the JTS connection had been used for a standard JDBC call. All standard calls established the underlying connection, but `getVendorConnection()` did not. The problem was corrected with a code fix.
CR127891	The format of the connection leak file was modified to make the information more readable.

CR131575, CR132652	WebLogic Server sometimes threw a bogus warning about a JDBC connection leak that began:
	
	`[SerialConnection] : Connection Leak`
	`detected!!!!!!java.lang.Throwable: StackTrace at creation of`
	`connection: /n`
	
	The leak detection code that sent this warning is obsolete. A code change resolved the problem.

jDriver

Change Request Number	Description
CR125135	By default, WebLogic jDriver for Oracle/XA Data Source set the value of the `oracleXATrace` parameter to true, rather than false. This caused the driver to create trace files of the form `xa_poolname*.trc` that could grow large over time, unless you specifically disabled trace files by setting `oracleXATrace=?false?` in `config.xml`. The code was modified to set the default value of `oracleXATrace` to false if no value is specified.
CR129220	WebLogic Server Oracle jDriver was not properly releasing Clob Objects for garbage collection. A code change resolved the problem.
CR113462	WebLogic Server threw a `NumberFormatException` when using `BigDecimal` 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.

JMS

Change Request Number	Description
CR135798	The messaging pipeline was not being managed correctly for transacted MDBs. Messages already in the pipeline were not bumped when new messages with higher priority arrived on the queue. Therefore, it was important to keep the message pipeline size to a minimum to get the desired behavior. The code was modified to decrement the window count on acknowledge from a transacted MDB. The messaging pipeline for transacted MDBs will honor the `MessagesMaximum` setting.
CR110120	If you migrated a JMS server from one Managed Server to another, sent some messages, and then migrated the JMS server back to the first Managed Server, you saw exceptions like the following: `<Jun 23, 2003 4:18:49 PM EDT> <Error> <JMS> <040366> <JMS Distributed Destination member "S2_BE1_T1" in "DT1" unable to accept connection from remote member "S1_BE1_T1" due to exception weblogic.jms.common.JMSException: Failed to setup system subscription because destination S2_BE1_T1 is not avaiable (shutdown, suspended or deleted). <Jun 23, 2003 4:19:54 PM EDT> <Error> <JMS> <040366> <JMS Distributed Destination member "S1_BE1_T1" in "DT1" unable to accept connection from remote member "S2_BE1_T1" due to exception weblogic.jms.common.JMSException: Consumer already exists` Analysis revealed that while JMS internally binds more multiple WebLogic Server Aggregatable objects on a single server instance, WebLogic Server JNDI expected only one Aggregatable per server and as a result was replacing the first JMS Aggregatable with the second. WebLogic Server JNDI now provides an internal API for JMS to handle this situation so all objects are preserved.
CR132606	Messages Pending counters in console were incorrect for Distributed Topic members that were on the same WebLogic Server instance. Counters were correct for remote distributed topic members. Code was changed to correctly managed the distributed topic reference count when forwarding to other non-remote distributed topic members. The customer will no longer see unexpected messages pending for distributed topic members after all messages have been successfully received and acknowledged.

Change Request Number	Description
CR128596	Message flows sometimes stalled when messages were being sent over the bridge using distributed destinations from a 7.0 SP2 cluster to 8.1 SP2 cluster when one of the 8.1 SP2 cluster servers was bounced.
	A code change resolved the problem.
CR127260	`MessagingBridgeRuntime` Means was not stopped by the `weblogic.Admin` utility command:
	`java weblogic.Admin -url t3://127.0.0.1:7001 -username weblogic -password weblogic INVOKE -type MessagingBridgeRuntime -method stop`
	The `MessagingBridge` start and stop methods now throw an informative exception about how to start and stop `MessagingBridge` at runtime.
CR126183	An idle bridge was logging a message after the maximum idle time setting had been reached.
	Code was added to suppress the repetitive log message "Bridge X start transferring messages" logged by an idle bridge.
	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.
CR125693, CR092468	The legal minimum value of the `flowMinimum` attribute of the `JMSConnectionFactoryMBean` had been changed to 1 and so, to be consistent, the legal minimum value of the `flowMaximum` attribute was also changed to 1.
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.
CR103001, CR127436	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.
CR107729, CR113714, CR124985	A problem involving the accumulation of expired messages in a destination with no consumer has been resolved with a code change that adds support for active message expiration. This feature allows expired messages to be removed from a destination that does not have a consumer. To turn on the active message expiration, use the system property `-Dweblogic.jms.ExpirationScanIntervalSecs="number of second(s) "` between scan. Zero and negative numbers will disable the feature.

Change Request Number	Description
CR108665	The JMS server sometimes deleted expired paged-out messages from the persistent store, regardless of whether a queue browser might still have it in the reference list. As a result, enumeration operations sometimes ended due to paging IO exception even when more messages might still be available in the enumeration list. A code change improved the handling of paging, and solved the problem.
CR111159	If JMS front-end and back-end are not co-located and the front-end `FEProducerSendRequest` gets a Failure/JMSException, later, when the `FEProducerFiniteStateMachine` is resumed in a different thread, it does not fail over. Analysis revealed that if the producer load balancer selects a destination member, but the back-end throws an exception when attempting to add the message to the destination, the front-end did not properly handle the exception and select another distributed destination member for the send. The problem was resolved by a code change to make sure the front-end producer catches the exception from the back-end and retries, selecting another available distributed destination member. If no other member is available the exception is returned to the sending application.
CR110991, CR117044	The `JMSServerRuntimeMBean` method `getHealthState` is no longer a public method.
CR112845	The JMS dispatcher was not able to handle a failover of a JMS send operation when the client and front end (the server where the JMS Connection Factory was targeted) were co-located and the JMS Distributed Destination Member was on a different VM. A code change has resolved this problem.

Change Request Number	Description
CR120619	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: `<Aug 13, 2003 4:50:58 PM PDT> <Error> <JMS> <Failed to deploy JMS Server "server_name" due to weblogic.jms.common.JMSException: Error initializing JMSServer server_name. weblogic.jms.common.JMSException: 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.addDeployment(DeploymentTarget.java:364) at weblogic.management.mbeans.custom.DeploymentTarget.addDeployment(DeploymentTarget.java:150) at java.lang.reflect.Method.invoke(Native Method)`  `[...]` 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.

Change Request Number	Description
CR121011, CR131700	A `NullPointerException` resulted on restart of JMSServer, when the store contained a message that had a ReplyTo that was a Distributed Destination and was not in the same cluster as the store.
	There were two WebLogic Server domains which used the MessagingBridge for inter-domain communication. The sending application set the JMSReplyTo field as a DistributedDestination on domain1, then the message traveled over the bridge and ended up in the JMSStore on domain2. Upon restart of the JMSServer in domain2, JMS attempts to locate the configMBeanName for the Distributed Destination member instance that was written to the store, but it cannot locate this name because it belongs to domain1. This is where the failure occurred. The log contained messages similar to:
	`####<Aug 7, 2003 1:36:13 PM BST> <Info> <JMS> <hwbachp> <managed5_hwbwl98> <main> <kernel identity> <> <040108> <User connection factory "BridgeJMSnnn" is started.> ####<Aug 7, 2003 1:36:13 PM BST> <Info> <JMS> <nnn> <managed5_nnnn> <main> <kernel identity> <> <040108> <User connection factory "Wnnn_JMSConnectionFactory" is started.> ####<Aug 7, 2003 1:36:13 PM BST> <Info> <JMS> <nnn> <managed5_nnnn> <main> <kernel identity> <> <040108> <User connection factory "nnnQueueFactory" is started.> ####<Aug 7, 2003 1:36:13 PM BST> <Info> <JMS> <hwbachp> <managed5_hwbwl98> <main> <kernel identity> <> <040108> <User connection factory "WLI_B2B_TopicFactory" is started.> ####<Aug 7, 2003 1:36:13 PM BST> <Info> <JMS> <hnnn> <managed5_nnnn> <main> <kernel identity> <> <040108> <User connection factory "RNQueueFactory" is started.>`
	The problem was resolved with a code modification, so that if JMS cannot locate the configMBeanName, it uses the instance name to create a DestinationImpl (non-DD queue) for the replyTo field. The replyTo Queue is downgraded from a DistributedDestination to a normal destination for this scenario.
	CHANGED BEHAVIOR: If the message is in the store on server restart a) which has a ReplyTo and b) this ReplyTo is not in the same cluster to which this store belongs and c) this replyTo is a Distributed Destination, then this replyTo will be plugged in as a normal Destination. Clients using this replyTo will not get LoadBalancing because the replyTo is downgraded from Distributed Destination to a normal destination because the configMBeanName cannot be found (from another cluster).

Change Request Number	Description
CR121041	A problem occurred with a Messaging Bridge between two JMS topics running in two Managed Servers in the same domain, running on the same machine. As the Messaging Bridge was being started during WebLogic Server startup, it threw the following warning message. Messages could not be forwarded.

```
####<Aug 20, 2003 12:49:11 PM MST> <Warning> <MessagingBridge>
<slsol4.bea.com> <server2> <ExecuteThread: '10' for queue: 'de
fault'> <kernel identity> <> <200026> <Bridge "Test Messaging
Bridge" encountered some problems to one of its adapters or und
erlying systems. It stopped transferring messages and will try
to reconnect to the adapters shortly. (Exception caught was ja
va.lang.Exception: javax.resource.spi.IllegalStateException:
Managed connection is closed at
weblogic.jms.adapter.JMSManagedConnection.checkIfDestroyed(JMS
ManagedConnection.java:263) at
weblogic.jms.adapter.JMSManagedConnection.sendEvent(JMSManaged
Connection.java:231) at
weblogic.jms.adapter.JMSBaseConnection.throwResourceException(
JMSBaseConnection.java:1266) at ...
```

Analysis revealed that the bridge failed to create the message listener because it was configured for durable topics and there was no JMS store available. The Bridge encountered an internal error when trying to log the resource exception so the customer was not able to tell why the bridge was failing.

The problem was resolved by a code change to allow the bridge to throw the correct resource exception. Now, the correct exception is logged by the bridge.

```
<Aug 27, 2003 9:32:35 AM CDT> <Warning> <MessagingBridge>
<200026> <Bridge "TopicBridge" encountered some problems to one
of its adapters or underlying systems. It stopped transferring
messages and will try to reconnect to the adapters shortly.
(Exception caught was java.lang.Exception:
javax.resource.ResourceException: Error setting message
listener . . .
. . .
-------------- Linked Exception ------------
javax.resource.ResourceException: Error creating asynchronous
consumer or setting message listener . . .
. . .
-------------- Linked Exception 2 ------------
weblogic.jms.common.JMSException: No store for durable consumer
. . .
```

Change Request Number	Description

CR121741	Inconsistent behavior occurred when shutting down the MQ-WLS bridge using MBeans. This excerpt is the code used for shutdown:
	
	``
	MessagingBridgeMBean bridge = (MessagingBridgeMBean)mbeanHome.
	getAdminMBean(bridgeName, "MessagingBridge");
	bridge.setStarted(false); for(int i=0;i<serverList.length;i++)
	{ target=(TargetMBean)
	mbeanHome.getAdminMBean(serverList[i],"Target");
	bridge.removeTarget(target);
	``
	
	Sometimes the bridge was untargetted and shut down with the following exception:
	
	``
	<Aug 27, 2003 9:24:08 AM PDT> <Info> <MessagingBridge> <200034>
	<Bridge "reverseBridge" is shut down.> <Aug 27, 2003 9:24:08 AM
	PDT> <Error> <Connector> <190008> <Error closing connection
	instance for XA Transaction Resource Adapter.>
	javax.resource.spi.IllegalStateException: Managed connection
	is closed at
	weblogic.jms.adapter.JMSManagedConnection.checkIfDestroyed
	(JMSManagedConnection.java:263) at
	weblogic.jms.adapter.JMSManagedConnection.cleanup(JMSManagedCo
	nnection.java:128) at
	weblogic.connector.common.internal.XATransResourceFactory.clea
	nUp (XATransResourceFactory.java:325) at ....
	``
	
	Analysis revealed that MessagingBridge.shutdown() called MessagingBridge.cleanup(), which then calls recover() on the source connection. The exception occurred because Recover() is not a valid operation on a transacted session.
	
	A code changes has been implemented to verify that a session is not transacted before calling recover().

Change Request Number	Description
CR121760	An `InvalidDestinationException` was received when using WebLogic Server Messaging Bridge to integrate WebMethods 6.0 JMS with WebLogic Server. When sending messages to WebLogic Server, WebMethods 6.0 requests an `ack`. The bridge sends `ack` of type `weblogic.jms.DestinationImpl` when WebMethods expected type `com.wm.broker.jms.Destination`. This exception resulted:

```
javax.jms.InvalidDestinationException: Destination must be of
class: com.wm.broker.jms.Destination at
com.wm.broker.jms.Message.setJMSDestination(Message.java:896)
at weblogic.jms.client.JMSProducer._send(JMSProducer.java:380)
at weblogic.jms.client.JMSProducer.send(JMSProducer.java:172)
at
weblogic.jms.adapter.JMSBaseConnection.sendInternal(JMSBaseCon
nection.java:5 71) at
weblogic.jms.adapter.JMSBaseConnection.send(JMSBaseConnection.
java:528) at
weblogic.jms.adapter.JMSConnectionHandle.send(JMSConnectionHan
dle.java:131) at java.lang.reflect.Method.invoke(Native
Method) at
weblogic.connector.common.internal.ConnectionWrapper.invoke(Co
nnectionWrappe r.java:101) at $Proxy115.send(Unknown Source) at
weblogic.jms.bridge.internal.MessagingBridge.onMessageInternal
(MessagingBrid ge.java:1160) at
weblogic.jms.bridge.internal.MessagingBridge.onMessage(Messagi
ngBridge.java: 1093) at
com.wm.broker.jms.MessageConsumer.deliverMessage(MessageConsum
er.java:682) at
com.wm.broker.jms.Session$MessageDeliveryThread.run(Session.ja
va:1431)
```

As WebLogic Server is providing bridge functionality between the two JMS implementations, the request is to convert message acknowledges to the correct type. It seems that this is correctly done for regular messages.

Analysis revealed that the standard WebLogic JMS client was handed a WebMethods message to send, and attempted to call `setJMSDestination` on the WebMethods message using a WebLogic destination.WebMethods throws an InvalidDestinationException when this occurs.

The problem was resolved by a code change to catch and ignore the InvalidDestinationException.

Change Request Number	Description
CR122749	Sorting on a `DestinationKey JMSCorrelationID` could have resulted in a `NullPointerException` if not all of the messages in the queue had a `JMSCorrelationID` set. The same failure could have occurred for `DestinationKey JMSType`.
	Code was modified to check for a null `JMSCorrelationID` and null JMSTypes before calling `compareTo`. As a result, sorting will work correctly, without throwing a `NullPointerException`.
CR123194	In previous WebLogic Server 7.0 service packs, when the server instance went down but its clients remained active, JMS threw a runtime exception `weblogic.rmi.extensions.RemoteRuntimeException`, instead of a JMSException as expected per the JMS specifications.
	A code change resolved the problem.

Change Request Number	Description
CR123675	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:

```
<Sep 12, 2003 9:54:12 PM EDT> <Error> <Kernel> <BEA-000802>
<ExecuteRequest
failed java.lang.NullPointerException.
java.lang.NullPointerException at
weblogic.jms.common.MessageImpl.writeExternal(MessageImpl.java
:1622) at
weblogic.jms.common.TextMessageImpl.writeExternal(TextMessageI
mpl.java:92) at
weblogic.jms.store.ObjectIOBypassImpl.writeObject(ObjectIOBypa
ssImpl.java:155) at
weblogic.jms.store.BufferDataOutputStream.writeObject(BufferDa
taOutputStream.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)
```

The problem was solved with a code fix.

CR128723, CR081630	Because of a timing issue with the consumer load balancer, enabling ServerAffinity caused behavior that could incorrectly select a remote member when a local member would be more appropriate.

In previous service packs, enabling ServerAffinity set this set of preferences:

1) Pick local member without a consumer

2) Pick local member

3) Pick remote member without a consumer

4) Pick remote member

As of Service Pack 5, enabling ServerAffinity sets this set of preferences:

1) Pick local member without consumer

2) Pick any member without consumer

3) Pick local member

4) Pick any member

JSP

Change Request Number	Description
CR172250	There was no backward compatible property for `weblogic.jspc`. Added a new command line flag for `weblogic.jspc` called `backwardcompatible`. If it is set to true, then the Web application will be compiled maintaining backward compatibility with WebLogic Server 6.x earlier. With the new "backwardcompatibility" flag, you can compile Web applications as in WebLogic Server 6.x.
CR133760	The server tried to read the Init parameters from the servlet config, using "compilerClass" String instead of "compilerclass" string. The JSP descriptor element is compilerclass and so the JSPServlet was changed accordingly. Code was added to change the method `config.getInitParameter("compilerClass")` to `config.getInitParameter("compilerclass")` in JSPServlet code. The JSP descriptor element "compilerclass" is now honored.
CR093625	The server sometimes misdirected output for JSP includes in JSP pages. Following a code change, `isResponseWrapper` is reset if all wrappers were removed (if, for example, all of the wrappers were JSP `NestedBodyResponse` wrappers from JSP body tags, since they are removed on a forward when a wrapper implements `weblogic.servlet.internal.RemoveWrapperOnForward`).
CR127836	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. A code change resolved the problem.
CR126007	When a JSP was forwarded from another JSP, the multi-byte string parameter was lost. This problem occurred only when the parameter encoding was EUC-JP or ISO2022JP and it was sent by the POST method. Code was added to prevent reparsing the post parameters if the encoding has not changed.
CR092039	WebLogic Server threw an `UnsupportedEncodingException` if you included extra quotation marks around the charset value of a JSP. For example, the following tag would yield the exception: `<%@ page contentType="text/html; charset=\"Shift_JIS\"" %>` The problem was resolved with a code fix.

Change Request Number	Description
CR093520	When you precompiled JSPs on a machine in one timezone, and then deployed those same JSPs on a server in a different timezone, 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. The problem was resolved by storing the timezone at compile time and using that timezone at deployment time to determine whether recompilation is necessary.

CR098514	Making a new line during string literal in an attribute of a JSP custom tag caused a compilation error. For example:
	
	`<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core";; %>`
	`<c:if test="${ sessionScope.sessionID == null`
	`  or sessionScope.dummy == null }">`
	`  There is no Session ID!`
	`</c:if>`
	
	Which led to the following error:
	
	`unclosed string literal`
	`        _c_if0.setTest(weblogic.utils.StringUtils.valueOf("${`
	`sessionScope.sessionID == null //[ /ifTest.jsp; Line: 17]`
	
	A code change resolved the problem by allowing the storing of new line characters in the attribute values of a JSP custom tag.

CR104429, CR134313	WebLogic Server was recompiling JSP pages after updated JSP class files were copied to `jspWorkingDir`. A new parameter, `-Dweblogic.jsp.alwaysCheckDisk`, when set to true causes WebLogic Server to check the stale JSP class from disk . This parameter is set to false by default, so default behavior is not changed.
CR111024, CR134046, CR111897	For a JSP with the following fragment using the `Cache` tag: `<wl:cache name="email_portlet" key="request.email_portlet_key" scope="session" timeout="60"> time=<%= new java.text.SimpleDateFormat("kk:mm:ss.SSS").format(new Date()) %> key=<%= key %> <% for (int i=0; i<100; i++) {// just a delay try { Thread.sleep(10); } catch (Exception e) { } } refreshed = true;%>`  and a concurrent load of ten users, ten threads deadlocked at `weblogic.cache.CacheSystem.waitOnLock()`. A code change resolved the problem.

Change Request Number	Description
CR111423, CR127336	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.
	The problem occurred because of a difference in the rounding behavior of timestamps used in the jar and zip 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.
	The code was modified to advance the timestamps in compiled JSP classes by one second, thereby preventing JSPs from being recompiled.
CR111655	Previously, it was not possible to use JavaServer Pages Standard Tag Library (**JSTL**) tags in JSPs that include Japanese characters. When such a JSP is executed, an error starting with the following lines occurred:
	`java.io.IOException: javax.servlet.jsp.JspException: The taglib validator rejected the page: "org.xml.sax.SAXParseException: An invalid XML character (Unicode: 0x82) was found in the CDATA section."`
	If the `pageEncoding` attribute was not specified in the page directive, the byte stream, which is used for taglib validation, was constructed using default encoding.
	The problem was corrected with a code change to ensure use of the character encoding defined in the `contentType` attribute if the `pageEncoding` is not specified.
CR112794	A JSP was refreshed with a copy that did not parse correctly, and this caused a deadlock.
	The problem was resolved by a code change that added a check during JSP refresh.
CR117477	For `response.sendRedirect`, the Content-Type was set to text/plain by default, instead of text/html. As a result, WebLogic Server returned a page with 503 status code for content that required the text/html Content-Type.
	Following a code change, text/html is the default type when the Context-Type is not defined yet by the time `sendRedirect` is activated.
	Note that a receiving party who is relying on the text/plain Content-Type will fail after this change.

Change Request Number	Description
CR120914	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.
CR123520, CR106226	In WebLogic Server 7.0 Service Pack 4, if the `PageContext.include()` or `forward()` method was used in a JSP body tag, and `HttpServletResponse.getOutputStream()` was called in the included or forwarded servlet, the following exception occurred: `java.lang.IllegalStateException: Cannot use ServletOutputStream because a Writer is being used. Use getWriter() instead.` This included request was probably nested in a JSP body tag. A code change has resolved the problem.

JTA

Change Request Number	Description
CR111475	`JTARuntime` MBean was incrementing statistic counts on non-coordinating servers, as follows: If you started a transaction, enlisted a resource from a Managed Server (making the Managed Server the coordinator), then enlisted a resource from the Administration Server, and committed the transaction, the `JTARuntime` MBean from both servers returned `TransactionTotalCount` as 1. Following a code change, WebLogic Server only updates statistics on the coordinating server.
CR122842	When registering a new resource under the same name as the old resource, any subsequent XA transaction operations could not go to the new resource. They continued to attempt to reach the original resource. Code was added to allow the substitution of a second resource with the same name.
CR127412	The application could not get a connection when `PinnedToThread` was turned on or when using the Oracle OCI NativeXA driver. The problem was resolved by a code change.

Change Request Number	Description

CR113226	When a resource name contained more than 64 characters, WebLogic Server could throw the following exception when testing the connection pool:
	
	`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) [...]`
	
	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.

CR134081	When a resource adapter becomes unregistered, either by calling `unregister` or by a resource failure, all transaction operations intended for that resource adapter fail. Starting with Service Pack 5, if a second resource adapter with the same name is registered, all transactions that were intended for the first resource adapter will now operate on the second resource adapter.

CR126201	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:
	
	`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`
	
	`[...]`
	
	The code was fixed to obtain new address information from the Administration Server in response to an address or port change.

Node Manager

Change Request Number	Description
CR104285, CR124729	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.
CR127930	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp.
CR125829	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_42.00.jsp.

Operations, Administration, and Management

Change Request Number	Description
CR099488	WebLogic Server was calculating the Administration port URL incorrectly. A code change ensures that the URL is calculated correctly, based first on the Administration Port (if it is enabled), and then on the Administration channel (if it is enabled).
CR128713	Users sometimes got a beanshell exception when setting the ErrorDestination on a JMSQueue. In certain circumstances the parent was not set on the JMSQueue and caused setting the ErrorDestination to fail because the legal check used the parent. Code was added to modify the legal checker to pull the parent JMSServer from the name of the JMSQueue for the check.

Change Request Number	Description
CR134167	EXISTS_POOL did not work as expected. It was looking for runtime mbeans which may or may not exist even if the pool exists. DELETE_POOL, an undocumented command, was the internal implementation to delete a connection pool; however, this feature was documented as DESTROY_POOL externally.
	The Help menu also displayed some undocumented commands.
	EXISTS_POOL command implementation now looks for configuration mbeans to check whether the pool exists.
	DESTROY_POOL command now uses the underlying DELETE_POOL implementation.
	All the undocumented commands that appeared in the `weblogic.Admin` help menu are now disabled. Commands include TEST_POOL, REMOVE_POOL, SUSPEND_POOL, SHUTDOWN_POOL, RESUME_POOL, DELETE_POOL. These will not be supported going forward and will not work in 7.0 SP5.
	The Help menu no longer lists these commands.
CR136210	The code attempted to do a `runAs` where parameters needed to be passed. Instead of passing the parameters on the stack, however, the code attempted to stash them into object variables. It also tried to use a token scheme to identify which of the log methods to call from `runAs()`. These object variables were not synchronized. As a result, the same message was sometimes logged multiple times.
	The code was changed to use an anonymous inner class which made the code much cleaner and more thread safe. The code in the log handler was synchronized to write to the log file.
CR130441	The security sample providers sample on dev2dev was broken in earlier service packs of 7.0. When you built the sample, started the server, and tried to set up the domain, you received a number of `javax.management.MBeanException` errors.
	A code change was made that enables the server to consider "commotype" as a flag to the administration tool.
CR101109	Upon enabling JDBC logging for a Managed Server in a cluster, the Console showed an error while starting. This error did not effect the JDBC logging and JDBC logging worked correctly.
	Code was added to fix the logging server debug messages on Managed Servers.
CR121228	User identity is now being recorded in auditing log messages.
CR121216	The @exclude and @non-configurable tags on the `DomainMBean.AdministrationMBeanAuditingEnabled` attribute have been removed.

Change Request Number	Description
CR132445	`weblogic.Admin` did not work properly with the default URL. For example: `java weblogic.Admin -username ss -password ss VERSION` returned no results, but adding `-url t3://localhost:7001` returned correct results. Following a code change, `weblogic.Admin` now checks the input URL for null, because it already has the default URL internally.

CR111199	Restarting the Administration Server, and then starting two Managed Servers simultaneously resulted in heavy CPU consumption. One Managed Server completed initialization, the second started initialization and stopped.
	
	Use of `kill -QUIT pid` revealed that the main thread was stuck in `HashMap.rehash`, as shown:
	
	`"main" prio=5 tid=0x29240 nid=0x1 runnable`
	`[0xffbec000..0xffbedbd4] at`
	`java.util.HashMap.rehash(HashMap.java:292) at`
	`java.util.HashMap.put(HashMap.java:344) at`
	`weblogic.management.internal.DynamicMBeanImpl$XInfo.get(Dynam`
	`icMBeanImpl.java:2270) at`
	`weblogic.management.internal.DynamicMBeanImpl.<init>(DynamicM`
	`BeanImpl.java:195) at`
	`weblogic.management.internal.DynamicMBeanImpl.<init>(DynamicM`
	`BeanImpl.java:167) at`
	`weblogic.management.runtime.RuntimeMBeanDelegate.<init>(Runti`
	`meMBeanDelegate.java:70) at`
	`weblogic.jms.frontend.FEConsumer.<init>(FEConsumer.java:100)`
	`at weblogic.jms.frontend.FESession$3.run(FESession.java:952)`
	`at ....`
	
	A code change increased the size of the default HashMap to accommodate all MBean attributes without resizing, and synchronized thread access to the HashMap.

Change Request Number	Description
CR111287	The Convert weblogic.properties utility did does not add DOCTYPE to `web.xml` and `weblogic.xml`.
	The utility was corrected to add these sentences to the beginning of `web.xml` and `weblogic.xml`, respectively.
	```<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd";> <!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application 7.0//EN" "http://www.bea.com/servers/wls700/dtd/weblogic700-web-jar.dtd";> CONFIGURATION: WebLogic Server 7.0 SP2```

Change Request Number	Description
CR120987	A Managed Server log was created, incorrectly, in the NodeManager home directory (NODEMGR_HOME) instead of in the RootDirectory specified. The access.log and the tlog files were correctly created under the RootDirectory.
	NodeManager had been configured on a machine, to allow remote start of the Managed Server.
	In the RemoteStart tab for the Managed Server in the console, the RootDirectory for the Managed Server and the FileName for the log were specified.
	config.xml contained this stanza for the Managed Server:

	<pre><Server ExpectedToRun="true" ListenAddress="172.23.64.45"
	ListenPort="27023" Machine="MyMachine" Name="m1"
	ServerVersion="7.0.3.0" StdoutSeverityLevel="64">
	<COM Name="m1"/>
	<ExecuteQueue Name="default" ThreadCount="15"/> <IIOP
	Name="m1"/>
	<JTAMigratableTarget Cluster="" Name="m1"
	UserPreferredServer="m1"/>
	<JTARecoveryService Name="m1"/>
	<KernelDebug Name="m1"/>
	<Log FileName="m1/m1.log" Name="m1"/>
	<SSL Name="m1"/> <ServerDebug Name="m1"/>
	<ServerStartName="m1"
	OutputFile="/home/support/BEA/wlserver7.0sp3/user_projects/
	jay1/./NodeManagerClientLogs/jay1_m1/startserver_106132100875
	4.log" Password="{3DES}7v6S0vZT+xUhhbWkLEq23A=="
	RootDirectory="/home/support/BEA/wlserver7.0sp3/user_projects
	/jay1" Username="system"/>
	<WebServer Name="m1"/>
	</Server></pre>

	The problem was corrected by a code change.
CR121839	DomainMBean instances did not reflect the auditing state when enabled from the command-line.
	If MBean auditing is enabled on the command-line via the -Dweblogic.AdministrationMBeanAuditingEnabled switch, neither the Domain nor the DomainConfig MBean instances showed the attribute being set to true. Auditing was on, but the Mbeans did not reflect that fact.
	The problem was corrected with a code fix to set the domain mbean attribute from the system property during server startup.

Change Request Number	Description
CR121728	Enhancement:
	Mbeans for custom authentication providers no longer must be put inside of the WebLogic installation tree in the `WL_HOME`\server\lib\mbeantypes directory.
	`WL_HOME`\server\lib\mbeantypes is the default directory for installing MBean types. However, if you want WebLogic Server to look for MBean types in *additional* directories, use the `-Dweblogic.alternateTypesDirectory=<dir>` command-line flag when starting your server, where `<dir>` is a comma-separated list of directory names. When you use this flag, WebLogic Server will always load MBean types from `WL_HOME`\server\lib\mbeantypes first, then will look in the additional directories and load all valid archives present in those directories (regardless of their extension). For example, if `–Dweblogic.alternateTypesDirectory = dirX,dirY`, WebLogic Server will first load MBean types from `WL_HOME`\server\lib\mbeantypes, then any valid archives present in `dirX` and `dirY`.
	Note that you must continue to use the option or your server will be come unbootable (for example, if you used the option and created some users and then decided not to use the `alternateTypesDirectory` option).
	For more information, see "Install the MBean Type Into the WebLogic Server Environment" in *Developing Security Providers for WebLogic Server*.
CR122204	When a Managed Server with a custom COMMO bean was restarted without restarting its Administration Server, the Managed Server was unable to create an instance of the COMMO bean. This exception occurred:
	`javax.Management.InstanceAlreadyExistsException is thrown, and the Detailed message in the exception is, The Object name specified 'wlidomain:Name=t,Server=managed1,Type=AppViewRuntime' is not unique across the domain. Please choose an unique Object Name`
	Analysis revealed that the Mbean was not un-registered on the Administration Server's MBeanServer when it went down. For this reason, upon restart, the Managed Server could not re-create the bean—the Administration Server still had a bean instance with that bean's name in its list of instantiated beans. Mbean names must be unique across a domain.
	The problem was resolved by a code change to un-register all of a Managed Server's server-specific beans when it goes down. When a Managed Server goes down, the Administration Server receives a `PeerGoneEvent` and un-registers all MBeans associated with the Managed Server.
CR136718	The configuration of "Root Directory" is now working when using Node Manager.

Plug-ins

Change Request Number	Description
CR134413	The Apache plug-in caused a duplicated http header and body for the 302 response. There was no problem between the plug-in and backend servers, but the Apache server added an additional 302 response.
	Code was added which reverted the return value of the request_handler method to OK.
CR113033	In earlier service packs, the ISAPI plug-in did not recognize the `WLTempDir` flag for the _wl_proxy folder. The code was fixed to use the flag.
CR132699	When the application servers are down the Apache access logs should record a 500 error but instead returned a 200 code:
	``` [Mon Jan 05 12:58:21 2004] [notice] Apache/2.0.44 (Unix) configured -- resuming normal operations ```
	``` [Mon Jan 05 12:58:25 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	``` [Mon Jan 05 12:58:27 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	``` [Mon Jan 05 12:58:29 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	``` [Mon Jan 05 12:58:31 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	``` [Mon Jan 05 12:58:33 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	``` [Mon Jan 05 12:58:35 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 ```
	The problem has been resolved.
CR130060	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.

Change Request Number	Description
CR129471	In previous service packs, the Apache plug-in did not recognize the `WLTempDir` parameter. This has been corrected.
CR129342	The ISAPI plug-in sent the `WL-PATH-TRIM HTTP` Header value to a WebLogic Server in place of the `WL-PATH-TRIM` value.
	A code change resolved the problem.
CR129138	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.
	A fix to cookie parsing and the substitution of JVMIDs to locate primary and secondary servers resolved the problem.
CR129026, CR129323	A memory leak in the ISAPI plug-in was fixed by a code change.
CR127973	The ISAPI plug-in sometimes failed after adding a persistent cookie to a servlet session.
	A correction to the cookie parsing code resolved the problem.
CR127658	If a connection is grabbed from the pool, but the server instance has already closed the connection, the `HALF_OPEN_SOCKET_RETRY` exception was thrown, causing the deletion of the previous connection object and the creation of a new one to connect to the same server.
	The problem was resolved by the addition of code to handle the `HALF_OPEN_SOCKET_RETRY` exception properly.
CR127231	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 READ_ERROR_FROM_SERVER or a CONNECTION_REFUSED exception was raised.
	Code was added which marks the server as bad on getting a 503 HTTP status error, gets the next available server and resends the request. All requests will now successfully fail over to next available server.
CR132840	Apache access logs improperly recorded a 200 code rather than a 500 error when application servers were down. A code change resolved the problem.

Change Request Number	Description
CR135002	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. A code change has resolved the problem.
CR112503	String comparison for headers was case sensitive for the NSAPI plug-in, and is now case insensitive.
CR123775, CR123120	Using the Post method to forward requests through the Apache plug-in caused the plug-in to set `content-length` to 0 and `wlproxy.log` to log this message: `POST and PUT requests *must* contain a Content-Length` This problem has been resolved with a code fix.
CR121726, CR121341	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_39.00.jsp.

CR110991,
CR117044,
CR120970,
CR120978
	Requests that received the PROTOCOL_ERROR message from the primary server also received a message similar to the following:
	
	`Mon Jun 30 21:52:57 2003 general list: trying connect to`
	`'10.84.133.182'/7305/7306 at line 1257 for ..`
	
	Indicating that failover was applying to another server in the general list, instead of the secondary server.
	
	The problem has been resolved with a code fix.

Change Request Number	Description
CR126982	When `WLExcludePathOrMimeType` was set, the file types were cut in the request to WebLogic Server, but iplanet failed to serve those files instead.
	For example, this request for a .jsp that contained a .jpg:
	``` <Object name="test5" ppath="*/weblogic/*"> Service fn="wl_proxy" WebLogicHost="lorna" WebLogicPort="7001" PathTrim="/weblogic" Debug="ALL" DebugConfigInfo="ON" WLExcludePathOrMimeType="*.jpg" </Object> ```
	was proxied to WebLogic Server, and Iplanet failed to serve the jpg. The Iplanet access log contained this message:
	``` 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)] ```
	`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.
	The problem was solved with a code change.
CR126568	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.
	The problem was corrected by code change to the plug-in to detect and handle HTTP/0.9 responses correctly.

Change Request Number	Description
CR126103	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 ns-httpd process crashed.
	The same load test did not crash on HP11.00 or Solaris.
	Analysis revealed a problem in the code in `proxy.cpp` used `strdup()`, a native system call that allocates system memory to the program's heap space. WebLogic Server uses Iplanet's FREE macro to free previous allotted space when it is no longer needed. Because FREE does not free the allocated space by the `strdup()` call, the memory leak occurred.
	The problem was solved by replacing all native `strdup()` system calls in `proxy.cpp` with Iplanet's STRDUP macro so the FREE macro is instructed what space to free.
CR125690	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:
	`Thu Oct 09 13:01:46 2003 *******Exception type [WRITE_ERROR_TO_CLIENT] raised at line 1269 of .\iisproxy.cpp`
	Diagnosis revealed that the `Reader::fill()` method was not allocating enough memory while growing the initial buffer. Four bytes to mark the end of buffer were getting lost, which resulted in the core dump. The problem was solved with a code fix.
CR125545	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. This problem was solved with a code fix.
CR124464	The plug-in could cause iPlanet to crash shortly after a request received a `HALF_OPEN_SOCKET_RETRY, "Unexpected EOF reading HTTP status - request retry"` message.
	The code was modified to retrieve the exception code from the exception object and then delete the object.
CR124433	If IIS was configured with `WlForwardPath=/`, the plug-in would try to 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.

Change Request Number	Description
CR123925	The plug-in sometimes responded to the browser with a 500 error message. This problem had three additional symptoms:

1. The IIS access log would show the message:

```
Out-of-process+ISAPI+extension+request+failed. 500 1726
99122 2003 84078
```

2. The Windows Event Log would record Event ID 37:

```
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.
```

3. The `wlproxy.log` entry showed:

```
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
```

This problem was solved with a code fix.

Change Request Number	Description
CR123120	If the POST method was used through the plug-in and the `Content-Length` was not defined, the proxy log file would contain message such as: `POST and PUT requests *must* contain a Content-Length` The code was modified to set a content length of zero (0) if `Content-Length` is undefined.
CR122755	The plug-in filter was bypassed if `?.wlforward?` was manually appended to a URL. The code was modified to throw a 404 error if the initial request has a mime type of `.wlforward`.
CR122754	The plug-in parameter `WLExludeByPathOrMimeType` did not work when forwarding by mime type. This problem was solved with a code fix.
CR122207	If `KeepAliveEnabled` and `DynamicServerList` were both enabled, the plug-in could leave sockets in a `CLOSE_WAIT` state. This problem was solved with a code fix.
CR121688, CR121943	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.

CR113093	When using multiple `MatchExpression` parameters in `httpd.conf` to route requests to different locations, as in:
	
	`MatchExpression *.jsp WebLogicHost=localhost|WebLogicPort=8001`
	`MatchExpression *.html`
	`WebLogicCluster=localhost:8001,localhost:8003`
	
	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 `*.jsp` requests going to the server at port 8003.
	
	The code was fixed to ensure that each request uses its own copy of the parameter information.

CR111167	In previous service packs of WebLogic Server 7.0, 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.
CR110664	The plug-in code failed to catch an exception, which in turn caused iPlanet server to crash during the `sendResponse` phase. The plug-in code was changed to catch the exception.

Change Request Number	Description
CR109755	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: `LocationMatch "/weblogic/(abc|def)/ghi"` This problem was solved with a code fix.
CR108092	In previous service packs of WebLogic Server 7.0, 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: The HTTP Server encountered an unhandled exception while processing the ISAPI Application. This problem was solved with a code fix to the ISAPI plug-in.
CR105123	If you configured a virtual directory, configured a mime type with the wildcard * (proxy everything), and added a `DefaultFileName` in the `iisproxy.ini` file, on a request for a directory with no filename the `DefaultFileName` was not used. A code change resolved the problem.
CR105173, CR135917	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 TO CLIENT ERROR] was inappropriately logged in the Web server logs. The server no longer logs a 500 error for this kind of response failure.
CR091910	In earlier WebLogic Server 7.0 service packs, the Apache plug-in did not read `PathPrepend` when using `<IfModule mod_weblogic.c>`. The problem occurred with plug-ins for Apache 1.3.x and Apache 2.0.43. This problem appeared only if `PathPrepend` and `MatchExpression` properties existed together within the `<IfModule mod_weblogic.c>` tag. This was also true for `PathTrim` property. The problem was solved with a code fix.
CR106764	A thread in the Netscape plug-in could obtain a critical lock for a long duration (5 minutes by default, or configured using `WLIOTimeoutSecs`), blocking all other threads and making the Web server appear to hang. This problem was fixed with a code change to the plug-in.
CR087204	Using the `PathTrim` and `PathPrepend` parameters in conjunction no longer creates a URL with an unexpected forward slash "/" at the end.

RMI

Change Request Number	Description
CR124596	An optional enhancement to the BEA ORB forces reconnection when bootstrapping and allows hardware load-balancers to correctly balance connection attempts.
	For more information on this feature and its limitations, see Using RMI over IIOP with a Hardware LoadBalancer in *Programming WebLogic RMI over IIOP*.
CR125261	A fat client application instantiated clients that called a stateless session EJB and experienced a memory leak when using WebLogic Server version 7.0 with Service Pack 3 and Service Pack 4.
	This issue was resolved by using the context classloader in the Appmanager. `-Dweblogic.LoadStubUsingContextClassLoader` option will no longer be available after this change.
CR127333	When an RMI/IIOP client was killed when the server was writing a response, some subsequent client requests failed with a `MarshalException`. When the initial client connection broke, the indirection map was not cleared when put back into the pool. Subsequent clients which used that particular map from the pool failed with the `MarshalException`.
	A code change ensures that the indirection maps are properly cleared even in cases when the client gets killed.
CR121079	When making an initial request, the WebLogic Server-IIOP runtime would not select the appropriate codeset for wide string transmission.
	The appropriate codeset for wide string transmission is now selected for all requests. Interoperability with WebLogic Server will now work correctly.
CR128594	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.

Change Request Number	Description
CR124377	WebLogic Server sometimes threw a `java.rmi.UnmarshalException` when a client application using the thin-client .jar (`wlclient.jar`) accessed an EJB. On the server, partial exception was:
	`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)[...]`
	On the client, the partial exception was:
	`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.mapSystemException(ShutdownUtilDelegate.java:97) at javax.rmi.CORBA.Util.mapSystemException(Util.java:65) [...]`
	This problem did not occur when using weblogic.jar on the client. The code was modified to address this problem.
CR109844, CR113715	Out Of Memory errors due to memory leakage of the `DGCServerHelper` class during repeated creation and close of `InitialContext` unnecessarily create a helper object to manage distributed garbage collection.
	A code change fixed the problem.

Change Request Number	Description
CR106281	In previous WebLogic Server 7.0 service packs, replicated session beans under heavy load resulted in increasing heap usage and `OutOfmemoryErrors`.
	Analysis revealed that `examples.ejb20.basic.statefulSession.TraderBean_5ysgq2_EOImpl` objects were not being garbage-collected even when garbage collection was forced. For clustered stateful session beans, strong references were maintained to the `EOImpl` object primary in the `weblogic.rmi.cluster.PrimarySecondaryRemoteObject`. As a remove is never called on the bean, the reference to the `EOImpl` was never removed from the `eoMap`.
	A code fix was implemented to unexport the EO when the passivated bean has been deleted, after `session-timeout-seconds`.
CR131692	When throwing a derived exception from the ejb method, the iiop layer was adding the Full Value Description for the derived exception but not for the base exception. When the client came back to get information about the base exception the server did not have it and threw an exception.
	Code has been added to ensure that when creating a Full Value Description for the derived class, the Full Value description of the base classes is created as well.

Security

Change Request Number	Description
CR135710	There was a `wait(1000)` in the `backendstandard` that was delaying the some operations such as undeploy.
	A code change removed the `wait(1000)` and introduced `tp.wait()`.
CR172774	The `CertificateAttribute` needs to be defined to match the binary field that holds the certificate in LDAP, whether it is `"userCertificate"` or `"userCertificate;binary"`, as described below. Currently it defaults to `"userCertificate"`.
	1. `userCertificate`
	If you load the certificate via the ldapbrowser, the attribute `userCertificate` is created, of type binary, whose value is the data of the certificate.
	To be able to access the certificate you must define `CertificateAttribute="userCertificate"`.
	2. `userCertificate;binary`
	If you use ldapmodify to add the new attribute, as in:
	`ldapmodify -p 1155 -D Principal -w Password`
	`dn: cn=support@bea.com,ou=Certs,dc=bea,dc=com`
	`changetype: modify`
	`add: userCertificate`
	`userCertificate;binary:: MIICxDCCAi2gAwIBAgIBIDANBgkqh.....`
	an attribute named `userCertificate;binary` is created where the certificate data is loaded. To be able to access this certificate you must define `CertificateAttribute="userCertificate;binary"`.
CR121114, CR134158, CR134159, CR135017, CR130506	The SocketImpl class did not catch an IOException and continued to rethrow it, causing the socket not to close.
	This problem was resolved by adding code to close the socket and throw the IOException. As a result of this change, there are no idle sockets.
CR120748	In prior releases, WebLogic Server did not have an LDAP X509Identity Asserter.
	Code was added to allow the use of an LDAP X509 Identity Asserter.

Change Request Number	Description
CR112147	The existing ServletAuthentication API methods such as weak, strong, authenticate etc, were not propagating the LoginException back to the caller. Two overridden login methods were added, which perform similarly to the weak and authenticate methods. The assertIdentity method, which performs in the same way as the strong method, was added. These new methods will propagate the LoginException to the calling code.
CR124746, CR175051, CR175045	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_48.00.jsp.
CR132503	The Web services client loaded the local identity directly into the Certicom SSL context, but the Web services `HttpsURLConnection` class didn't associate the connection with that specific Certicom SSL context. The connection is now associated with the SSL context by the `HttpsURLConnection` class, which takes an SSL context as a parameter.
CR130355	When adding a duplicate entry, `DefaultCredentialMapperLDAPHelper` threw an exception without completing the method even though the exception was harmless. A code change causes `DefaultCredentialMapperLDAPHelper` to ignore the exception while adding duplicate entries.
CR128002	The connection pool was not being maintained correctly, and the size of the pool was insufficient for the load the request generated. These problems caused a performance problem while authenticating users. A code change corrected the problems with pool. In addition, an option was added to set the LDAP connection pool size. The option is `-Dweblogic.security.providers.authentication.LDAPDelegatePoolSize`.
CR127259	The month field in the `DefaultAuditRecorder` backup file name was reporting an incorrect value. Code was added to fix the expression. The Month field now reports the correct value.
CR125409	When two Managed Servers were started simultaneously, then same code was executed twice to create an LDAP entry as a part of deploying an application. The lack of synchronization was causing the creation of a duplicate entry in the LDAP. Code was added to check whether an entry exists in the synchronization before updating the id counter. If a duplicate entry is being created accidentally, WebLogic Server throws an exception, which prevents the creation of the entry.

Change Request Number	Description
CR064593, CR121607	The following command-line arguments used to control the SSL time-to-live and cache size are no longer ignored in WebLogic Server 7.0: `-Dweblogic.security.SSL.sessionCache.size=`*`sessioncachesize`* `-Dweblogic.security.SSL.sessionCache.ttl=`*`sessioncachetimetolive`*
CR097734, CR090472	`SSLLayeredSocket` was not creating `SSLIOContext` when proxying requests, causing `NullPointerException` while tunneling SSL via proxy servers. A code change that creates `SSLIOContext` and adds it to context table solved the problem.
CR105933	WebLogic Server was unable to filter connections from LDAP servers because the LDAP protocol was not allowed in the connection filter rules. LDAP has been added to the list of filterable protocols, so that connections from `ldap` can be filtered.
CR106192	When you set the security debug flags `<ServerDebug DebugSecurityAtn="true" DebugSecurityAtz="true" ...>` with `StdoutDebugEnabled="true" StdoutSeverityLevel="64"`, the server logged the user password to stdout in clear text. This problem has been resolved by removing the password string from the statement that is written to the log file.
CR106294, CR132596, CR120273	Removal of an application was delayed by a configured wait in the processing of embedded LDAP transactions. A code change has resolved the problem.

Change Request Number	Description
CR106532	When two-way SSL is turned on, the server invokes `IdentityAssertion` automatically, which is correct in a client-server scenario but causes problems when the connection is between instances of WebLogic Server. The scenario:
	Client hits WebLogic Server Instance1 and is authenticated correctly with two-way SSL. Instance1 then invokes an EJB on Instance2 via two-way SSL, and Instance2 authenticates based on Instance1's certificate. There are two issues:
	1. The true user is the client, not Instance1, but Instance2 reads the user as Instance1 based on the DN in Instance1's certificate.
	2. Authentication should not be required. You should be able to enforce client certificates for the SSL connection, but turn off the `IdentityAssertion` check if you want to use a trusted WebLogic Server domain.
	A code change has made certificate authentication configurable. If you do not want to treat the certificates submitted as a part of two-way SSL to do `IdentityAssertion`, use `-Dweblogic.security.disableIdentityAssertion=true` in the server startup.
CR107363	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_40.00.jsp.
CR108021	A problem that caused loss of information stored with the security subject has been resolved as follows.
	A new cache is used when a user logs in via two-way certificates or a token in a header or a cookie, improving the performance of identity assertion. The server receives an RMI call either from a 6.x or earlier instance of WebLogic Server, or when a servlet or EJB with a run-as tag is executed. If a user is deleted, identity assertion will continue to work for the deleted user until the cache TTL is reached. The cache TTL defaults to five minutes and can be adjusted on the command line as follows:
	`-Dweblogic.security.identityAssertionTTL=N` for

	`N > 0     sets the cache TTL to N seconds`
	`N = 0     sets the cache TTL to infinity`
	`N < 0     disables the cache altogether`

CR108624, CR128228	In a performance tuning enhancement, new attributes were added that allow you to limit the depth of a group membership search. You can tune the search according to your membership hierarchy, and eliminate searching that you know will not find group members.
	The new attributes are `GroupMembershipSearching` and `MaxGroupMembershipSearchLevel`.

Change Request Number	Description
CR110242	The `LDAPConnection` used for the `realm.getGroups()` call retained the results and prevented their return to the pool, which was also used by an iteration of `getGroups()`. This caused a proliferation of open LDAP connections.
	Following a code change, an LDAP connection is associated with the `getGroups()` call to the group iterator, and the connection is closed when there are no more elements left in the iterator.
CR110977	Specifying `java.lang. Integer.MAX_VALUE` for `ACLCacheTTLPositive` resulted in a `java.lang.IllegalArgumentException`. The problem occurred after this sequence of events.
	1. Start WebLogic Server in `CompatibilityRealm` mode.
	2. Under the Caching realm, enable the ACL cache and specify the value of ACL Cache Positive TTL = java.lang.Integer.MAX_VALUE and save changes.
	3. Restart WebLogic Server.
	WebLogic Server throws the following exception:

	<pre><Jun 25, 2003 4:31:44 PM PDT> <Notice> <Management> <140005>
	<Loading configuration
	D:\opt\weblogicapps\ecommerce\.\config.xml> <Jun 25, 2003
	4:31:47 PM PDT> <Info> <Logging> <000000> <FileLogger Opened at
	D:\opt\weblogicapps\ecommerce\.\logs\ecommerce\myserver.log>
	<Jun 25, 2003 4:31:50 PM PDT> <Critical> <WebLogicServer>
	<000364> <Server failed during initialization.
	Exception:java.lang.IllegalArgumentException: ttl<= 0
	java.lang.IllegalArgumentException: ttl <= 0 at
	weblogic.security.acl.TTLCache.<init>(TTLCache.java:195) at
	weblogic.security.acl.TTLCache.<init>(TTLCache.java:173) at
	weblogic.security.acl.CachingRealm.<init>(CachingRealm.java:47
	8) at
	weblogic.security.acl.CachingRealm.<init>(CachingRealm.java:37
	5) at
	weblogic.security.acl.CachingRealm.<init>(CachingRealm.java:35
	0) at ...</pre>

	The problem was solved with a change to the `ttlcache` creation code.
CR111305	The `webflowCheckAccess()` method now explicitly checks whether a logged in client has access to Web application resources before rendering them.

Change Request Number	Description
CR112971, CR104667	The NetScape LDAP API released with earlier service packs of WebLogic Server 7.0 contained a bug that could cause deadlocks when more than one thread called `deregister()` on `LDAPConnThread`. A code change resolved the problem.
CR113459	In previous service packs, removing the Node Manager properties file caused problems. The Node Manager properties file is always created in the `<saved logs dir>/NodeManagerInternal` directory. Periodically archiving and deleting the contents of NodeManagerInternal removed the Node Manager properties file, so that the certificate password stored in the Node Manager properties file could not be decrypted, and Node Manager did not work correctly. A code change creates the Node Manager properties file in the directory specified by `user.dir`, if the file is not found in `NodeManagerInternal`, or in the `user.dir` directory.
CR120233	A code change has resolved the problem that caused a `java.lang.ArrayStoreException` to be thrown when configuring a custom auditor and a custom role mapper with the IPlanet 5.1 Authenticator.

Change Request Number	Description
CR120850	`weblogic.net.http.HttpsURLConnection` did not honor `https.nonProxyHosts` environment variable. Requests from the client, whether one running within WebLogic Server or a standalone Java program, always went through the proxy even if the targeted host was specified in `https.nonProxyHosts`. `https.nonProxyHosts` is detected by `SSLParams` and an internal list of hosts that should be proxied is constructed. When the `SSLSocket` is created and initialized, the host is compared against this list. If the host appears on the list, the host is used to open the socket. If the host is not on the list, the `proxyHost` is used to open the socket. (And, if there is no proxyHost, then this is all moot and the host is used to open the socket). A code change resolved the problem. A user can now set the following properties: • `http.proxyHost` • `http.proxyPort` • `http.nonProxyHosts` • `https.proxyHost` • `https.proxyPort` • `https.nonProxyHosts` (new property) For either `http` or `https`, one can define a `proxyHost` upon which the socket will be opened. However, one can also define a list of hosts (`nonProxyHosts`) to which a client should be connected directly, even if a `proxyHost` is defined. The change made by this fix is to add `https.nonProxyHosts`, so that a user can now specify the hosts that the client should always connect to directly using SSL even if `https.proxyHost` is defined.

Change Request Number	Description

CR120852	A `MalformedURLException` resulted when the dynamic group attribute `memberURL` contained a string value with a reserved character (such as forward slash '/') in the LDAP search filter portion of the LDAP URL. For example, this URL:
	
	`ldap://`*`text replaced`*`:389/dc=`*`text`*
	`replaced``2??sub?(&(|(objectclass=person)`
	`objectclass=groupofuniquenames))(uid=*`*`text replaced`*`/2*))`
	
	resulted in this exception:
	
	`<Jun 23, 2003 12:30:35 PM EDT> <Debug> <SecurityDebug> <000000>`
	`<advance dyn group entry = LDAPEntry: cn=slashgroup,ou=Groups,`
	`dc=`*`text replaced`*`; LDAPAttributeSet: LDAPAttribute {type='cn',`
	`values='dynamicgroup001,slashgroup'} LDAPAttribute`
	`{type='memberURL', values='ldap:///dc=`*`text`*
	`replaced``??sub?(&(|(objectclass=person)(objectclass=groupofuniq`
	`uenames))(uid=*h`*`text replaced`*`/2*))'}> <Jun 23, 2003 12:30:35 PM`
	`EDT> <Debug> <SecurityDebug> <000000> <advance member url =`
	`ldap:///dc=`*`text`* `...`
	
	Analysis and investigation revealed that:
	
	• In an LDAP URL, '/' is not illegal, although it is a URL delimiter and must be escaped.
	• WebLogic Server used the `LDAPUrl` object to get the filter, while iterating the dynamic groups. If that url filter part contains special characters like '/' then the `LDAPUrl` class is not handling them properly.
	
	The problem was resolved with a code change to obtain the filter without using `LDAPRealm` object.

CR120932	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. A code change corrected the problem.
CR121043, CR134995	Properties set directly in the JSP were not being captured and set in the HTTP client. A code change has resolved the problem.
CR121135, CR123865	JDK 1.3.1_08 threw an internal exception that caused two-way SSL communication between two WebLogic Server instances to fail under certain circumstances when `LANG` was set to `univ.utf8`. A code change has resolved the problem.

Change Request Number	Description
CR121646	A `BAD_CERTIFICATE` error was received and the SSL connection was terminated when a client certificate with Extended Key Usage set to `critical` was sent to WebLogic Server. Analysis revealed that WebLogic Server did not support Enhanced KeyUsage in SSL. The problem was resolved by adding support for EnhancedKeyUsage. WebLogic Server can now accept Certificates with Enhanced Key Usage set to `critical`.
CR122642	After an upgrade from Service Pack 2 to Service Pack 3, an attempt to view the Security->User->Realms->myrealm->Users screen resulted in WebLogic Administration Console hang. A code change has resolved the problem in Service Pack 5.
CR124164, CR090738,	The entitlement engine previously restricted the number of operands in a role expression to 100. This restriction in turn limited various attribute values. For example, the maximum number of `principal-name` elements that could be mapped to a `role` within a `security-role-assignment` in `weblogic-ejb-jar.xml` was limited to 50. Having more than `principal-name` 50 elements resulted in a `weblogic.entitlement.data.EnCreateException` during the deployment of an application. A code change resolved the problem by removing the restriction on the number of operands in a role expression.
CR126106, CR090472	On deployment of an application, LDAP entries are written first to Administration and then to Managed Servers. In previous service packs, if writing to the Administration Server failed, the call was returned without writing to the Managed Server, which caused the EJB security layer not to return the correct roles. A change to the LDAP code has resolved the problem.
CR133655, CR128002	Connection pool code was not behaving properly, causing the LDAP searches to slow down, which caused, for example, slow authentication with a third-party LDAP. A code change resolved the problem.

Servlets

Change Request Number	Description
CR128624, CR171907	When an application attempted to encode a URL which included an HTML anchor tag, such as this: http://localhost:7001/WebApp/target.jsp#section4, the resulting URL was incorrect: http://localhost:7001/WebApp/target.jsp#section4;jsessionid= 16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDLl5urAhK96dqvfo!-1835542719.
	The anchor tag should be at the end of the URL, so it is ignored. The correct URL would be: http://localhost:7001/WebApp/target.jsp;jsessionid= 16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDLl5urAhK96dqvfo!-1835542719#section4.
	Code was added to look for an anchor if there is no query string in the original URL, and to append the anchor to the end of the encoded URL.
CR123333	Form-based authentication in a Web application did not allow multi-byte characters for 'j_username' and 'j_password', which prevented request encoding in JSPs. Multi-byte characters caused authentication to fail.
	To specify character encoding in the JSPs, you can use 'j_character_encoding' as follows:
	```\n<form method="POST" action="j_security_check">\n<input type="text" name="j_username">\n<input type="password" name="j_password">\n<input type="hidden" name="j_character_encoding"\nvalue="Shift_JIS" >\n</form>\n```
	Note: j_character_encoding is not J2EE compliant.
	A code change has resolved the problem.
CR132522	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:
	```\nHTTP/1.1 404 Not Found\nDate: Thu, 01 Jan 1970 00:00:00 GMT\n```
	This header is not valid according to section 14.18 of RFC2616. A code change resolved the problem.
CR129237	A servlet that invoked HttpServletRequest.getServletPath() or HttpServletRequest.getPathInfo() received incorrect values when the ServletMapping was for *.<value> and the URL contained a full stop '.'
	Fixing a parsing error resolved the problem.

Change Request Number	Description
CR129211	Using `ServletOutputStream.write(byte)` to write more than the buffer size caused infinite loop. A code change resolved the problem by updating the check for boundary conditions when the buffer is full and autoflush is set to false.
CR128986	After a protocol exception, the server hung while trying to process a new license because of a problem with the `ensureContentLength` method. A code change resolved the problem.
CR128420	`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. A code change resolved the problem.
CR127708	`HttpParsing.unescape()` decoded the path `/foo/..%c0%../` to `/foo/..` because `sun.io.ByteToCharUTF8.convert()` stops converting the remaining bytes if it encounters bytes that are not valid in UTF8 (for example, `0xc0`). This problem did not occur on JDK 1.4. The problem was resolved with a WebLogic Server UTF-8 converter that replaces invalid UTF-8 sequences with U+FFFD characters. As a result of this fix, `HttpParsing.unescape()` decodes the path `/foo/..%c0%../` to `/foo/..?%..`
CR125846	According to the Servlet Specification, an exact pattern should take precedence over a wildcard pattern. But this was not working correctly. For example if you have: "/TestPath/*" maps to "WildcardServlet" and "/TestPath" maps to "ExactMatchServlet", if the incoming relative URI is '/TestPath' then `ExactMatchServlet` should have been served, not `WildcardServlet`. This was fixed by making appropriate changes in the pattern matcher
CR125048	Debugging information from `server/bin/fastfile.dll` was printed out in the Administration Console. Rebuilding the binary without any source code change resolved the problem.
CR133558	`ServletContext.setAttribute` threw a `NullPointerException` when a null value was passed to it. The problem was resolved by a code change that calls `removeAttribute()` when a null value is passed to the `setAttribute()`.

Change Request Number	Description
CR084455	When WebLogic Server was configured to log all HTTP requests in extended format, at the rotation time, the following error was thrown at approximately one-minute intervals after the rotation was scheduled (one minute being the log flush time interval setting): ``` ####<Aug 20, 2002 9:00:00 PM PDT> <Error> <HTTP myapp> <host2> <myserver> <ExecuteThread: '6' for queue: 'default'> <system> <> <000000> <Exception flushing HTTP log file> java.io.IOException: Failed to rename log file on attempt to rotate logs at weblogic.servlet.logging.LogManagerHttp.rotateLog(LogManagerHt tp.java:168) at weblogic.servlet.logging.LogManagerHttp.access$2(LogManagerHtt p.java:148) at weblogic.servlet.logging.LogManagerHttp$RotateLogTrigger.trigg er(LogManagerHttp.java:432) at weblogic.time.common.internal.ScheduledTrigger.executeLocally(ScheduledTrigger.java:238) at weblogic.time.common.internal.ScheduledTrigger.execute(Schedul edTrigger.java:229) at weblogic.time.server.ScheduledTrigger.execute(ScheduledTrigger .java:69) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) ####<Aug 20, 2002 9:00:05 PM PDT> <Error> <HTTP myapp> <host2> <myserver> <ExecuteThread: '7' for queue: 'default'> <system> <> <000000> <Exception flushing HTTP log file>... ``` Analysis revealed that the log had been flushed inappropriately. The problem was solved with a code change that ensures that the log is not flushed on rotation, and that the log file name is checked for a null value.
CR088785, CR122315	In HTTP access logs, WebLogic Server recorded only the stem portion of the URI, rather than both the stem and query portions, when the cs-uri field was specified. This problem was solved with a code fix.

Change Request Number	Description
CR092625, CR112910	WebLogic Server threw a `NullPointerException` 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: ```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)``` The problem was solved with a code fix.
CR095945	The `CGIServlet` extracts the CGI scripts in a WAR Web application so that it can execute them. Previously, the servlet called these scripts without setting the current working directory. This meant that subscripts could not be called properly. A code change now ensures that the current working directory is set so that CGI scripts can call subscripts in WAR Web applications.
CR096459	When invoking a flush on 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.
CR103339	Servlet output capitalized the Transfer-Encode type value, Chunked, instead of using lowercase chunked as required by the HTTP/1.1 specification. The code was fixed to ensure that Servlets use the Transfer-Encode type "chunked" in lowercase.
CR103482, CR120943, CR129041	A workaround that solved the problem of an orphaned session in a secondary server by invalidating the session created a race condition that caused a "primary not found" error during load test. A code change resolved the problem.
CR103925, CR125206	The `setAttribute` method only checked for `hashCode` equality. It now also checks the result of the equals method for the old and new objects.

Change Request Number	Description
CR104245	Under heavy load, when both the primary and the secondary server in a two-server configuration access the same replicated session, a session replication deadlock results. Both servers try to replicate the same session data; the server in primary role locks the `ReplicatedSessionData` and tries to update the secondary server. At the same time, the other server does the same and waits for the first server to do the update of its secondary copy. The replication call comes to the first server on its replication thread queue and it cannot finish the update until it can get lock on the session data that is held by another thread in the same server acting as a primary for the same session data. Each server keeps trying to replicate as primary and update the same session as secondary, leading to a mutual deadlock.
	As the servers deadlock, all requests involving session access and replication are blocked while the requests keep piling up. As the two servers keep toggling between primary and secondary state, "Got stale replication request" is logged.
	A code change that substituted a synchronized object for a synchronized method resolved the problem.
CR108034	Inappropriate error messages generated by a user breaking a connection have been suppressed.
CR108350	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.
CR108385	Attempts to post chunkpost data to cluster instances using `HttpClusterServlet` which was acting as a proxy in front of the WebLogic Server cluster generated the `NumberFormatException`, while direct requests without proxying from `HttpClusterServlet` succeeded without any errors.
	Taking into account that chunk transfer encoded data is already decoded by the core server, a code change that prevents `HttpClusterServlet` from making an unnecessary attempt to decode the data again fixed the problem.
CR108577	Module order was not considered during undeployment. This logic is fixed. Now the modules are deactivated and rolled back in the exact reverse order in which they were deployed.
CR108607	WebLogic Server produced errors when applications used XSL with the Formatting Objects Processor 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.

Change Request Number	Description
CR109479, CR110838	The `JnlpDownloadServlet` uses the timestamp on a WAR file to determine whether the client's resources are older than what is available on the Web server. To get the timestamp, `JnlpDownloadServlet` calls the `URLConnection.getLastModified()` method. However, the `getLastModified()` method returned 0. The code has been modified so that a `ZipURLConnection.getLastModified()` method returns the modification time of the zip file, and thus the WAR file.
CR109885	Session replication to a failed secondary node caused the primary node to hang. The primary node appeared to be attempting to connect to the secondary node and holding a lock that other threads had to wait on. A code change has resolved the problem.
CR109958	When processing requests through a proxy servlet, WebLogic Server only honored the `SecureProxy` setting for incoming requests that used HTTPS. If the incoming request used HTTP, WebLogic Server did not use an HTTPS connection even when `SecureProxy` was enabled in the proxy servlet. This problem was solved with a code fix.
CR110798	Calling `weblogic.servlet.security.ServletAuthentication.killCookie(req)` on 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 `killCookie(req)` completes.
CR110832	WebLogic Server threw a `NullPointerException` upon deployment of a Web application when `DebugEnabled` was set to `true` for the `WebAppComponent` in `config.xml`. The problem was resolved with a code change.
CR110910	`HttpClusterServlet` threw a `NumberFormatException` when `KeepAliveEnabled` was set to `true` after a large file download was canceled. 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.
CR110914	If `TrackingEnabled` was set to false, WebLogic Server created a new session for each request, but the sessions were not getting invalidated. The code was modified to invalidate a session immediately if `TrackingEnabled` is set to false. This is the correct behavior.

Change Request Number	Description
CR111090, CR090886	When `-Dweblogic.webservice.transport.http.full-url` was set to True, a Web service could not be invoked using the HTTPS protocol. A code change has resolved the problem.
CR111531	Weblogic Server 7.0 did not handle file names that contained double byte characters when the `Sending UTF-8 Only` option was turned off. Attempts to access a file with double byte characters in its name resulted in a 404 file not found error. The problem was corrected with a code fix.

CR111752	Under certain conditions, WebLogic Server threw a `NullPointerException` when using `CGIServlet` with the `useByteStream` parameter set to true. The problem occurred when using framesets where one frame contained static URL links and another frame used `CGIServlet`. 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:
	
	`java.lang.NullPointerException`
	`at`
	`weblogic.utils.Executable$Drainer.run(Executable.java:366)`
	
	This problem was solved with a code fix.

CR112799	WebLogic Server attempted to write to an output stream even after an `IOException` occurred. This led to 100% CPU utilization if an unexpected socket disconnection occurred with a Web Application that did not handle `IOException`. `Org.apache.xml.serialize.XMLSerializer` ignores `IOException` until the end of its process, which caused a problem when an `IOException` occurred in the middle of returning XML documents as part of an HTTP response. The code was modified to ensure that writes to an output stream stop after an `IOException` occurs.

Change Request Number	Description
CR112910, CR092625	WebLogic Server threw a `NullPointerException` when HTTP logging was enabled 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:
	``` java.lang.NullPointerException at weblogic.servlet.logging.LogManagerHttp.log(LogManagerHttp.jav a:292) at weblogic.servlet.internal.HttpServer.log(HttpServer.java:865) at weblogic.servlet.internal.ServletResponseImpl.send(ServletResp onseImpl.java:1044) at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRe questImpl.java:2265) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) ``` The problem was solved with a code fix.
CR114936	WebLogic Server drained an input stream even if a client cancelled a proceeding request, thus consuming CPU resources. The code was changed so that connections are thrown away so they do not drain an inputstream. As a result of this change, if an `IOException` occurs while writing to a client, `GenericProxyServer` does not recycle this connection and will create a new connection for a subsequent request.
CR116209	If customers added a serializable `ServletContext` attribute, then overwrote it with a non-serializable value, the original value masked the new value. This problem occurred because the old value was never removed from the attribute's Hashtable. The code has been modified to remove the value from an attribute's Hashtable, if necessary, when replacing a serializable value with a non-serializable one.
CR120228, CR124369	An exploded Enterprise application that contains exploded EJBs and Web applications and their MANIFEST files, which have shared classes in the classpath, worked on WebLogic Server 7.0 SP2 but not on WebLogic Server 7.0 SP3. The shared classes could not be found. This problem was resolved by a code change that adds the classpath entry from the MANIFEST file to the classloader for exploded applications.

Change Request Number	Description
CR120281	`HttpServletRequest.getParameterValues(String)` sometimes returned parameter values twice. Analysis revealed that query parameters in forwarded requests were being checked for parsing in a manner that caused properly parsed parameter values to be returned twice.
	A code change has resolved the problem.
CR120440	When multiple Web applications were deployed in a Single Sign-On configuration and one application called `weblogic.servlet.security.ServletAuthentication.invalidateAll(request)`, the `HttpSessionListeners` 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.
	The code was fixed to ensure that both the session ID and context path of all Web Application are registered for invalidation as necessary by `invalidateAll(request)`.
CR121094, CR129364	Redeploying a Web application that was deployed to both a cluster and a VirtualHost that also mapped to the cluster caused an `InstanceNotFoundException`. The problem was solved by a code change.
CR121175	`HTTPClusterServlet` routed requests to a different server instance than the primary instance identified by its JVMID in the session cookie.
	`HTTPClusterServlet` was proxying requests to multiple unclustered Managed Servers. The webapp on the backend server instance consisted an index.jsp and a frameset.jsp. The starting point of the webapp is index.jsp, where a session is created. `index.jsp` forwards requests using <jsp:forward> to frameset.jsp, which contains a frameset.
	The first time `index.jsp` is accessed by a proxy server (`HTTPClusterServlet`), a session is created and a cookie is set in response. When `index.jsp` forwards to frameset.jsp, new requests (for each JSP in frameset) are sent with the cookie. Intermittently, a cookie is found in the request but the request is sent to a server in server list (other the primary server) and the session is lost. The proxy log has the following entry:
	``` <Thu Aug 21 15:41:15 PDT 2003>: Found cookie: -1339390245 <Thu Aug 21 15:41:15 PDT 2003>: In-bound headers: <Thu Aug 21 15:41:15 PDT 2003>: Content-Length: 117 <Thu Aug 21 15:41:15 PDT 2003>: #### Trying to connect with server -1189081773!172.17.26.74!7201!443 ```
	The problem was solved with a change to the logic that updates the server list. When updating a JVMID for the server in the list, now the server is removed from list, updated, and then added back to the list. Simply updating the object did not sort the list.

Change Request Number	Description
CR121343, CR133921, CR129538, CR129537, CR174336	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.

CR121359	A `weblogic.utils.ParsingException` occurred when a JSTL end-tag contained a white space between the tag name and the right angle bracket(>). ex. </c:if >:
	
	`<%@ taglib uri="/WEB-INF/c.tld" prefix="c" %> Full stack trace:`
	`<Aug 25, 2003 12:46:23 PM EDT> <Error> <HTTP> <101017>`
	`<[ServletContext(id=53171`
	`81,name=TempApp,context-path=/TempApp)] Root cause of`
	`ServletException weblogic.utils.ParsingException: Could not`
	`complete parsing, unmatched tags: if otherwise when choose if`
	`forEach if at`
	`weblogic.servlet.jsp.JspLexer.parse(JspLexer.java:976) at`
	`weblogic.servlet.jsp.JspParser.doit(JspParser.java:90) at`
	`weblogic.servlet.jsp.JspParser.parse(JspParser.java:213) at`
	`weblogic.servlet.jsp.Jsp2Java.outputs(Jsp2Java.java:119) at`
	`weblogic.utils.compiler.CodeGenerator.generate(CodeGenerator.j`
	`ava:258 ) at`
	`weblogic.servlet.jsp.JspStub.compilePage(JspStub.java:353) at`
	`weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:211)`
	`at`
	`weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:164)`
	`at`
	`weblogic.servlet.internal.ServletStubImpl.getServlet(ServletSt`
	`ubImpl. java:517) at`
	`weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servle`
	`tStubIm pl.java:351) at`
	`weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servle`
	`tStubIm pl.java:306) .`
	
	The problem was resolved with a code change to allow zero or one occurrence of White Space between an tag's name and '>'.

Change Request Number	Description
CR121846	In previous WebLogic Server 7.0 service packs, 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.
	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.
CR122177	The `FileServlet` returned a response code of 200 instead of 404 when a file was not found. The code was fixed to return 404 when a file is not found.
CR122551, CR122556, CR129248, CR134888	`ServletOutputStreamImpl` threw the following exception for tunneled requests when the content length was set to -1 and written bytes were 0:
	`Servlet failed with IOException.java.net.ProtocolException: Didn't meet stated Content-Length, wrote: '0' bytes instead of stated: '-1' bytes.`
	A code change has resolved the problem.
CR124524, CR127621	The value `maxKeepAliveSecs` for `weblogic.management.configuration.WebServerMBean` has been made configurable up to a maximum of 300 seconds.
CR127888, CR111024	A server hung under load while using the cache filter. Analysis revealed that the problem involved unbalanced locked keys with respect to locked scopes.
	A code change resolved the problem.
CR134414	In previous service packs, if you added a serializable servlet request attribute and then overwrote it with a non-serializable value, the original value masked the new one.
	A code change has resolved the problem.

Simple Network Management Protocol (SNMP)

Change Request Number	Description
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.
	A code change has resolved the problem.

Web Applications

Change Request Number	Description	Release Fixed
CR175651	**Issue:** During undeployment or redeployment of Web applications, the container was not waiting for inflight Web application requests to be completed. As a result, if a Web application relied on using the context, it encountered an NullPointerException after undeployment, since the context was no longer valid. **Resolution:** As a fix, the -D option (namely, weblogic.http.requestCompletionTimeoutSecs) was introduced in the startup script file. The value given to this flag indicates the number of seconds for the container to wait for finishing inflight work. The default value is 0 seconds; therefore, the container does not wait if this flag is not present.	7.0 SP4

WebLogic Tuxedo Connector

Change Request Number	Description
CR125533	When CLASSPATH does not include an EJB JAR file, the invocation of a session bean's service method triggers object replication logic which results in a call to `TypedFML._tmpresend`. If the CARRAY field is null, the `_tmpresend` records the field length as 0 (instead of FLDID_SIZE + FLDLEN_SIZE). Ultimately, `_tmpostrecv` is called and assumes the field length is FLDID_SIZE + FLDLEN_SIZE. Because `_tmpresend` did not record this information, a negative value was read as field length. Currently the check made for field length is to check if it is 0. This is the reason for the `NegativeArraySize` exception. Code was added in `_tmpostrecv` to check the field length and determine whether it is less than or equal to 0.
CR074356	Multiple tBridge redirects could not be distinguished from each other because there was no Name attribute associated with a redirect. tBridge redirects now have a configurable Name attribute.

Change Request Number	Description
CR110812	Tuxedo Connector Queuing Bridge wrote debug messages to the WebLogic Server log even when the WebLogic Tuxedo debug option was not set. At intervals specified by the Timeout value in the Administration Console Tuxedo Queuing Bridge Connections tab page, debug messages were logged. For example, when Timeout was set to 60 seconds these debug messages were logged.

```
####<25-Jun-03 12:49:30 BST> <Debug> <WTC> <jdunn01>
<adminserver> <Thread-11> <kernel identity>
<59:223361a20602d183> <180046>
</tBexec/tuxQ2jmsQ/TPException
TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0>
####<25-Jun-03 12:50:30 BST> <Debug> <WTC> <jdunn01>
<adminserver> <Thread-11> <kernel identity>
<60:223361a20602d183> <180046>
</tBexec/tuxQ2jmsQ/TPException
TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0>
####<25-Jun-03 12:51:31 BST> <Debug> <WTC> <jdunn01>
<adminserver> <Thread-11> <kernel identity>
<61:223361a20602d183> <180046>
</tBexec/tuxQ2jmsQ/TPException
TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0>

......
```

The problem was solved by a code change.

| CR121355 | The reply message `TypedCArray` buffer was incorrectly padded with 4-bytes aligned—no padding should occur for a `TypedCArray` object. |

In a test of a tuxedo client making `tpcalls` to a wtc service using CARRAY message buffer, these results occurred:

- if wtc service send sback 174 bytes, the tuxedo client receives 176 bytes
- if wtc service sends back 175 bytes, the tuxedo client receives 176 bytes
- if wtc service send backs 176 bytes, the tuxedo client receives 176 bytes
- if wtc service send back s177 bytes, the tuxedo client receives 180 bytes

The problem did not occur when the service was running on another tuxedo domain gateway.

The problem was resolved with a code change.

Change Request Number	Description
CR121432	In the `weblogic.wtc.tbridge.tBexec` class, while running `tpenqueue`, the code was checking for null `replyQ` and replacing it if found with the name of the configured request queue. This check has been eliminated, and NULL is now a configurable value for `replyQ`.
CR122953	The `Fchg(FmlKey key, Object value)` method in the `weblogic.wtc.jatmi.TypedFML32` class did not correctly fill all prior entries with null values as documented. This problem was resolved with a code fix.

Web Services

Change Request Number	Description
CR122845	There was no way to create the `java.xml.soap.Detail` object of the `SOAPFaultException` exception in version 7.0, as the SOAP Attachments API for Java (SAAJ) was not implemented. This problem was resolved through a code change which exposed the WebLogic API `weblogic.webservice.util.FaultUtil`. This code change includes the `newDetail()` method for creating a Detail object. Note: the `weblogic.webservice.util.FaultUtil` API is deprecated in Versions 8.1 and beyond of WebLogic Server. You should use the SAAJ APIs in these versions to create the Detail object.
CR120594	Custom type mapping file specified by `typeMappingFile` in clientgen was ignored. This problem was resolved by setting the custom type mapping file and overriding the default mapping file.
CR103985	Setting the property in `"Setting javax.xml.soap.MessageFactory"` in the `startWebLogic.cmd` script did not work properly for `org.xml.sax.driver`, `org.xml.sax.parser`, `javax.xml.soap.MessageFactory` and `javax.xml.rpc.ServiceFactory` for JSPs and Servlets. Following a code change, WebLogic Server checks whether the property is already set before setting it, and sets it to the default value if is not set.

Change Request Number	Description
CR129061	The Apache SOAP client connects correctly to WebLogic Server Web services for methods that take parameters, but for methods without parameters the following error was generated on the server side: `javax.xml.soap.SOAPException: Found SOAPElement [''].  But was not able to find a Part that is registered with this Message which corresponds to this SOAPElement` A code change resolved this problem.
CR126896	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_47.00.jsp.
CR089677	The class `weblogic.xml.schema.binding.FaceUtils` was present in `webservices.jar` but not in `webserviceclient.jar`. Its absence generated an error when it was declared missing when a client passed complex parameters to a Web server (a list of Java objects, for example).
CR090950, CR172603	Accessing `SOAPElement.getElementName()` in a client-side handler caused a `ClassCastException` because the `SOAPElement` was created with a name class that did not implement `javax.xml.soap.Name`. The name class is now converted into `javax.xml.soap.Name`.
CR091230	The clientgen Ant task was removing the underscore "_" character when it generated Java code from WSDLs. A code change has corrected this problem.
CR097344	The autotype code generator did not properly handle attributes without types or with anonymous types. A code fix has resolved the problem.
CR099339	When you start the server with `-Dweblogic.webservice.verbose=true`, verbose output shows the response XML two times instead of the request and then the response XML. The title of the first output is "Request" even though it shows the response. Following a code change, the SOAP request and response print correctly to `stdout` on the server side when `Dweblogic.webservice.verbose` is turned on.
CR105715	The WebLogic Server `clientgen` Ant task was incorrectly allowing hyphens to remain in WSDL files. This caused generated Java code to have class and method names that contained hyphens, which is not legal in java. A modification to `NameUtils` now causes `clientgen` to strip out hyphens. Additionally, for JAXRPC methods and classes, if the resulting string is also a Java keyword, WebLogic Server prepend a _ to it, as per JAXRPC Specifications 1.0 and 1.1

Change Request Number	Description
CR106741	Specifying useServerTypes with multiple services in clientgen did not work properly. If you ran servicegen with multiple service entries to generate a client that specified useServerTypes=true, the type files were not always copied from the Enterprise application for the client, and the generation of the type code removed the "_" from the method names for the getters and setters.

You can now specify more than one service in a servicegen task whose client sets useServerTypes=true. For example, the following will now work:

```
<target name="ear">
    <servicegen
      destEar="RetailServicesServerBEA"
      warName="RetailServicesServerBEAWebApp">
      <service
javaClassComponents="com.hp.wsmo.demo.retail.webservice.
products.ProductsWS"targetNamespace="http://localhost:70
01/javaclass";
             serviceName="Products"
             serviceURI="/Products"
             expandMethods="true">
             <client
packageName="com.hp.wsmo.demo.retail.client.bea.
products" clientJarName="PRODUCTS.jar"
                useServerTypes="true" />
      </service>
      <service
javaClassComponents="com.hp.wsmo.demo.retail.webservice.
orders.OrdersWS"targetNamespace="http://localhost:7001/
javaclass";
             serviceName="Orders"
             serviceURI="/Orders"
             expandMethods="true"
             style="rpc"> <client
packageName="com.hp.wsmo.demo.retail.client.bea.orders"
                clientJarName="ORDERS.jar"
                useServerTypes="true" />
      </service>
```

Change Request Number	Description
CR107542, CR136469	In previous 7.0 service packs, only the URL path was used in the Request-Line when writing the request to the socket, and HTTP query and reference parameters in the Web service URI were not being passed to the Web service. Query and reference parameters, when present, are now appended to the request URI.
CR111879	When testing a Web service that used Date as an argument, the Web service displayed a security dialog requesting a username and password. The problem did not occur when the argument was changed to a string. The problem was solved with a code change to eliminate the security dialog and invoke the service correctly when it uses a Date argument.
CR112443, CR109898, CR126134	Passing `org.w3c.dom.Document[]` as a parameter to a method causes a `SOAPException` beginning with the following lines: `javax.xml.soap.SOAPException: Found SOAPElement [<anyType> <this xmlns="mynamespace"> <f:that xmlns:f="yournamespace"> <or> a lot of random < </or> <f:the> </f:the> <f:other> foo bar blaz</f:other> </f:that> </this> </anyType>]. But was not able to find a Part that is registered with this Message which corresponds to this SOAPElement ...` A code fix resolved this problem.
CR112940	In Service Pack 5, WebLogic Server implemented the following previously missing methods for the `SOAPHeader` class: • examineHeaderElements • extractHeaderElements • examineMustUnderstandHeaderElements • examineAllHeaderElements • extractAllHeaderElements
CR120076	A bogus Authorization header in a request caused the Web service engine to try to perform basic authentication even though there were no security constraints defined for the Web application. A code change has resolved the problem.

Change Request Number	Description

CR120548	In WebLogic Server 7.0 SP02 and SP03, when a user accessed the `webservices.basic.statelessSession` example Webservice using the client call:
	
	`String returnVal = port.sayHello(4, "\n\n <--spaces and`
	`\\n\\n");`
	
	the leading white spaces (new lines and spaces) were stripped off and did not appear on the server side.
	
	The problem was resolved with a code fix.

CR120741	Problems occurred when char was passed to a Web service operation, whether the client was static, dynamic, or the Web Service Test Home page. Passing char as a parameter resulted in the following assertion error:
	
	`run:`
	`     [java] weblogic.utils.AssertionError: *****`
	`ASSERTION FAILED *****[ invalid class: class`
	`java.lang.String ]`
	`     [java] at`
	`weblogic.utils.Debug.assertion(Debug.java:84)`
	`     [java] at`
	`weblogic.xml.schema.binding.internal.builtin.JavaCharSer`
	`ializer.getContentFromObject(JavaCharSerializer.java:23)`
	`     [java] at`
	`weblogic.xml.schema.binding.internal.builtin.`
	`XSDSimpleTypeSerializer.writeContentToStream(XSDSimple`
	`TypeSerializer.java:137)`
	`     [java] at`
	`weblogic.xml.schema.binding.internal.builtin.XSDSimpleTy`
	`peSerializer.writeContents(XSDSimpleTypeSerializer.java:`
	`130)`
	`     [java] at`
	`weblogic.xml.schema.binding.CodecBase.serializeInitialWr`
	`ite(CodecBase.java:370)....`
	
	The problem was resolved with a code change.

Change Request Number	Description
CR121394, CR128988	Calling a method in the Web service through the ISAPI filter caused this exception: `java.lang.IllegalArgumentException: Illegal MimeHeader name or value` A code change has resolved the problem.
CR122716	Clientgen failed to compile generated classes for an extended type if the base type had a different namespace. Analysis revealed that the base class from a different package was not being imported. The base class name is now generated with the complete package name, resolving the problem.

WebLogic Server 7.0 Service Pack 4 Solutions

The following sections describe problems resolved for the release of WebLogic Server 7.0 Service Pack 4 (SP4). The following list of resolved problems is updated periodically.

Administration Console

Change Request Number	Description
CR091853	For a three-node cluster on separate Solaris 2.8 machines, the Administration Server threw an InstanceNotFound exception when accessing the default Execute Queue.
	The problem has been resolved.
CR104066	The Applications Poller sometimes caused high CPU utilization.
	The `adminMBeanHome.getMBeansByType("Application")` now runs only once per poll interval.
CR105456	Selecting the Details tab in the Security->Realms->CompatibilityRealm->Providers->Adjudicators->RealmAdapterAdjudicator node in the Administration Console while using Compatibility Security caused a `javax.management.AttributeNotFoundException`.
	The Details tab was searching for an attribute, `RequireUnanimousPermit`, which is no longer specified for the `RealmAdapterAdjudicator` MBean. This problem has been resolved.
CR106122	WebLogic Server 7.0 previously required specification of the NAME attribute of the MLET element.
	To comply with JMX specifications, the NAME attribute is now optional.
CR110224	In a cluster environment, if the initial context was obtained from a Managed Server to access Mbeans using `getMBeansByType`, restarting the Administration Server resulted in an exception that began:
	`java.rmi.ConnectException: This RJVM has already been shutdown`
	Rebinding JNDI on Managed Server reconnect resolved the problem.
CR110557	A user belonging only to the Monitor Group was unable to monitor deployments, server states, and clusters. The problem has been resolved.

Cluster

Change Request Number	Description
CR108127	Switching from primary server A to secondary server B caused server B to be re-registered as the primary server without re-registering A as secondary. Switching from server A to server B and then back to server A resulted in a stale session.
	The primary status of server A is now removed when server B becomes the primary server.

Connector

Change Request Number	Description
CR106960	When a RAR was part of an EAR, redeployment of the EAR failed because the Connector Module code did not handle redeployment properly.
	Fixing the redeploy logic in the Connector module resolved the problem.

EJB

Change Request Number	Description
CR106041	`ejbc` now runs a compliance check that disallows optimistic concurrency for BMP (bean-managed persistence) beans.
	Optimistic concurrency is a feature of CMP (container-managed persistence) beans in EJB 2.0. See EJB Concurrency Strategy in The WebLogic Server EJB Container and Supported Services.
CR103391	The JDBC driver from Microsoft has a limitation whereby for a given result set of rows and columns, the getXXX method can only be called once per row. This limitation applied only if the query columns included a text or image column. WebLogic Server-generated JDBC obtained the key value from a row—for example, using a getLong(1)—and then passed the result set to another routine that read all the column values, including the first, re-executing a getLong(1). This re-execution caused the Microsoft driver to throw an exception.
	The problem is resolved by a code fix that avoids parsing the primary key columns in the resultSet twice.

Change Request Number	Description
CR105966	The inclusion of CMR fields in the `field-group` entry in `weblogic-cmp-rdbms-jar.xml` caused a compilation failure with the following error:
	`D:\AA\bea70sp1\weblogic700\samples\server\src\examples\ej b20\relationships\Fathr\EJB_Problem2>java weblogic.ejbc fatherError.jar fathersonError.jar ejbcgen\temp_lr_sahre_jb8\SonBean_1saa__WebLogic_CMP_RDBM S.java:909:`
	`incompatible types found   : <null>`
	`required: int __WL_bean.__WL_isLoaded[null] = true;`
	`1 error`
	`Exec failed .. exiting`
	The problem has been resolved.
CR106774	WebLogic Server-generated EJB SQL queries used ">= AND <=" as an operator. IBM DB2 does not handle "<=", and therefore requires "BETWEEN", which is equivalent.
	For DB2, generated EJB SQL now uses BETWEEN instead of ">= AND <=", and also handles NOT BETWEEN appropriately.

JDBC

Change Request Number	Description
CR109941	The creation of a localDataSource object from information specified in `weblogic-application.xml` resulted in a NullPointerException.
	The method `weblogic.jdbc.common.internal.LocalDataSource.defineDrive rProps`, was not checking for null values for `PoolParamsMBean`, `ConnectionCheckParamsMBean`, and `XaParamsMBean`.
	The problem has been resolved.

JMS

Change Request Number	Description
CR108665	The JMS server deleted an expired message from the persistent store when a queue browser still had the message in its reference list, resulting in a paging IO exception.
	A code change improved the handling of paging and resolved the problem.
CR109599	Creating a JMS JDBC store using Microsoft SQLServer 2000 caused an exception if messages were in the queue and you restarted the server.
	A change to the order of query and column retrieval resolved the problem.

JSPs and Servlets

Change Request Number	Description
CR103304	`jsp_precompile` now works for all kinds of JSPs, including those that extend custom classes.

Change Request Number	Description
CR096041, CR108111	WebLogic Server no longer uses redirection (HTTP status code 302) when returning a Welcome file.
	WebLogic Server versions prior to 8.1 SP2 and 7.0SP4 were returning a 302 (moved temporary) status code and the location of the Welcome file.
	WebLogic Server versions 8.1 SP2, 7.0 SP4 and later return a 200 (OK) status code displaying the content of the Welcome file.
	WebLogic Server performance is improved as a result of this change, however the relative URL in the Welcome file may point to a different location which would result in a 404-Not Found error.
CR107419	JSP code that intended to set double byte characters as a parameter's value—for example:

```
<jsp:include page="included.jsp">
    <jsp:param name="title"
     value="<%= java.net.URLEncoder.encode(\"[double byte
characters]\")
%>"/>
  </jsp:include>
```

—resulted in specified double byte characters being changed to their URL-encoded code. This problem has been resolved.

JTA

Change Request Number	Description
CR111719	The attribute `KeepXAConnTillTxComplete`, which is required for XA connection pools, now only performs a check if the XA isolation level has changed.

Node Manager

Change Request Number	Description
CR105924	Node Manager was unable to shut down a Managed Server after the Administration Server restarted.
	Node Manager now correctly updates the status of the Managed Server during shutdown.

Security

Change Request Number	Description
CR105443	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEA03_36.00.jsp.
CR110892	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-35.jsp.
CR112886	The identity assertion naming convention (WL-PROXY-CLIENT-*) for headers conflicted with WL-PROXY-CLIENT-KEYSIZE, WL-PROXY-SECRET-KEYSIZE and WL-PROXY-CLIENT-IP.
	Filtering these headers as identity assertion headers avoided a Client-Cert identity assertion error.

Tools

Change Request Number	Description
CR112500	The `weblogic.Admin` command did not return the correct exit code when the user and password provided did not authenticate. For example, for a non-existent user:
	`java weblogic.Admin -username george -password hamilton -url`
	`t3://localhost:7001 lock`
	This example returned an error message and stack trace indicating that the user did not authenticate, but the exit code indicated success.
	This problem has been resolved.

Web Services

Change Request Number	Description
CR104199	If you used a .NET C# client to access a WebLogic Server Web service and requested two strings returned, the first a result and the second an in/out parameter, the order of the strings returned was incorrect. The second string returned as the result, and the in/out parameter returned as null. A code change ensures that the first accessor returned is the return value.
CR092268	A client running behind a firewall needed to set properties on a Web service stub.
	The problem was resolved by the introduction of two new properties, `weblogic.webservice.client.proxyusername` and `weblogic.webservice.client.proxypassword`.
CR100098	A dynamic client for an SSL-protected Web service did not work when the port number was not included in the endpoint URL.
	The default SSL listen port, 443, is now automatically specified.
CR103512	Some properties set on the stub were not copied into the MessageContext.
CR107171	After a code change, WebLogic Server now supports the following usecases:
	Endpoint can be modified in the following flow:
	• `Stub->MessageContext` in `handleRequest->MessageContext` in `handleResponse->StubCall->MessageContext` in `handleRequest->Call`
	Timeout can be set in the following order:
	• Stub->MessageContext in handleRequest
	• Call->MessageContext in handleRequest
CR108579	A SOAP response from a document-literal Web service with arrays or user-defined types contained improperly namespace-qualified elements, causing .Net to ignore some data.
	The problem was resolved by passing namespace information to those SOAP elements.
CR109150	Using portable stubs created with the `weblogic.webservice.tools.versioning.VersionMaker` utility caused a `NoClassDefFoundError` exception.
	`wsclient70.jar` contained the server-side message logger instead of the client-side message logger. An update to `wsclient70.jar` resolved the problem.

Change Request Number	Description
CR110168	A client created by the clientgen task for a service that returns an array failed when the XML returned from the server did not explicitly indicate the size of the array.
	For SOAP arrays of unspecified size, objects are now stored in lists before the array is created.
CR110384	A missing `codecDir` in `wspackage` caused a `NullPointerException`.
	`codecDir` is not required in `wspackage`.
	The problem has been resolved.
CR111187	In the `weblogic.webservice.binding.soap.HttpResponse` class, the method `getBodyAsString` threw a "String index out of range" error.
	A code change causes the method to use the length of the body, instead of the end, resolving the problem.

WebLogic Server 7.0 Service Pack 3 Solutions

The following sections describe problems that were resolved for the release of WebLogic Server 7.0 Service Pack 3.

- "Administration Console" on page 4-152
- "Cluster" on page 4-157
- "Connector" on page 4-160
- "Core Server" on page 4-163
- "Deployment" on page 4-173
- "Domain Configuration Wizard" on page 4-174
- "EJB" on page 4-175
- "JDBC" on page 4-190
- "jDriver" on page 4-196
- "JMS" on page 4-197
- "JNDI" on page 4-208
- "JSPs and Servlets" on page 4-209

- "JTA" on page 4-222

- "Plug-ins" on page 4-226

- "RMI/RMI-IIOP" on page 4-229

- "Security" on page 4-232

- "System Administration" on page 4-238

- "Tools" on page 4-246

- "Web Services" on page 4-251

- "WebLogic Tuxedo Connector" on page 4-257

- "XML" on page 4-258

Administration Console

Change Request Number	Description
CR069284	You no longer configure the Auto Deployed Enabled attribute on the Domain --> Configuration --> Applications tab in the Administration Console. To control auto deployment behavior, see Auto Deployment in *Developing WebLogic Server Applications*.
CR069927	Selecting Server --> Monitoring --> Process Output and selecting any of the view options did not display the output. If you selected the "View Node Manager Output" while the Node Manager was running, the Node Manager would crash. A code fix resolved the problem.
CR076953	If you created a Role using the context-sensitive (right-click) menu item "Define Policies and Roles for individual beans..." and added the Role to the Policy (defined using the same context sensitive menu/form sequence) the desired resource could not be used by an external JCom client. The context-sensitive menu items create a scoped role, and clients outside the scoped role could not use the resource. Adding a "Define JCom Roles" menu item to the JCom node solved this problem by allowing revision of the scope to include the current client.

Change Request Number	Description
CR078764	Security settings in the Administration Console for Web services did not function properly.
	A code change removed the ability to set policy and roles for Web services in the Administration Console.
CR081673	Realms created by the Administration Console lacked the `UserLockoutManager` on their RealmMBeans.
	The Administration Console now creates a ULMbean when it creates a new realm.
CR081673	When customers configured a new security realm, the User Lockout tab displayed the message: 'User lockout feature not supported by the configured providers'. This problem resolved by modifying the code to ensure that a User Lockout Manager is created whenever a new security realm is created.
CR082335	The cursors created by the Administration Console for listing users and groups caused potential memory leaks when they did not close.
	Following a code fix, the cursors close.
CR083530	A new attribute, `LogTimeInGMT`, was added to the Server -- > Logging --> HTTP console page. This attribute specifies that the time stamp of HTTP log messages is written in Greenwich Mean Time regardless of whether the host computer has specified a different local time zone.
CR088262	Fixed a problem saving migratable targets for JMS servers where the targets reverted to an earlier setting.
CR089652	Disabling the SSL Port for a Managed Server from the Administration Console caused an `AssertionError` that begins:
	`weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[ ServerIron for server not found ]`
	Having the validation process check for the configuration MBean on the Managed Server fixed the problem.

Change Request Number	Description
CR090059	The WebLogic Server Administration Console requires you to specify how the WebLogic Security Service should perform security checks. You specify this preference using the `fullyDelegateAuthorization` flag, a command-line argument that you set when you start WebLogic Server.
	When the value of the `fullyDelegateAuthorization` flag is false, the WebLogic Security Service only performs security checks on URL and EJB resources that have security specified in their associated deployment descriptors (DDs). This is the default setting.
	See Understanding the fullyDelegateAuthorization Flag in *Securing WebLogic Resources*.
CR090266	The `Application` tag of `config.xml` has a `LoadOrder` attribute that was not settable from the Administration Console.
	A code change added the setting to the module pages.
CR090266	You could not set the `LoadOrder` attribute in the `Application` tag in `config.xml` from the Administration Console.
	Adding the `LoadOrder` attribute to application modules (`Application`, `ConnectorComponent`, `EJBComponent`, `WebAppComponent`, `JDBCPoolComponent`) fixed this problem.
CR090481	In previous Service Packs, when WebLogic Server was installed on NTFS volumes, you could not view the server log in the Administration Console if an alternate log file location was specified. With a log location setting like the following:
	<Server Name="myserver" ServerVersion="7.0.0.0">
	<Log FileName="c:\temp\myserver.log" Name="myserver"/>
	</Server>
	the server would throw a `java.io.FileNotFoundException`.
	This problem was solved by moving the call for the `FileName` attribute to a custom getter (breaking a previous dependency on `FileStreamHandler`), and having the `LogFileSearchServlet` get the `FileName` attribute directly from the MBean.
CR091170	When you tried to clone a new Web application, EJB, or connector on the Examples server, the Administration Console displayed an Unexpected Error page with this message:
	`javax.management.InvalidAttributeValueException: Path is NULL`
	The resolution was to remove the Clone icon from `ApplicationTable`, `EJBComponentTable`, `WebAppComponentTable`, `ConnectorComponentTable`.

Change Request Number	Description
CR091325	You can now configure policies for JMS topics and queues via the Administration Console.
CR091359	The Administration Console option Web Application->*webapp1*->Monitoring->Monitor all Active Web Applications... was failing for a Web application inside an EAR file. The problem was resolved with a code fix.
CR093289	Fixed a problem whereby log messages displayed in the Administration Console were truncating part of the text when using UTF8 encoding.
CR095556	The Administration Console threw the following `AssertionError` when you specified 00:00 as the format date for the Rotation Time for the access log: `weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[ Could not parse LogRotationTimeBegin: 00:00 because Unparseable date: "00:00" ]` Adding legal validation for the time format in the Log and WebServer MBeans fixed the problem.
CR095694	Fixed a display problem in the file browser associated with the `weblogic.properties` file converter, in Japanese.
CR098353	The TransactionLogFileWritePolicy attribute was added in 7.0 SP01, but was never exposed in the Administration Console. A field for setting this attribute was added to the Server —> Logging —> JTA tab.
CR099721	After edits to Cookie Max Age Secs using the Administration Console, the value sometimes reverted to the value before the edit. A code change removing a lower limit to the value fixed the problem.
CR102308	The `waitersTotalCount` value was not being decremented, causing the Administration Console to report incorrect values. Adding a new method, `WaiterCurrentCount`, resolved the problem.
CR102982	The Client Cert Proxy Enabled check box was displayed on the Tuning tab for a server, but this attribute is not related to tuning. The ClientCertProxyEnabled field was relocated to the Configuration —> General tab for a server.

Change Request Number	Description
CR103104	The `UserReader MBean` did not display users in the Administration Console. An authentication provider can read users in the Console while implementing the `UserEditor` but with `UserReader`, the message when trying to display users is "There are no Authentication providers available that support the creation of Users." The same thing was happening for Groups with the `GroupEditor` and `GroupReader` interfaces.
	The Administration Console was still calling the `userEditor.CreateUser` method, and never getting to the `UserReader.listUsers` method.
	The MBean has been fixed to look for appropriate provider type.
CR105624	The WebLogic Server Administration Console was accessible via the Managed Server address and port when the server had been started in MSI (Managed Server Independence) mode.
	A code change has restricted this inappropriate access.
CRO68646	The `weblogic.Admin` utility used the T3 protocol regardless of which protocol you specified. For example, the command "`java weblogic.Admin - url http://localhost:7001`" used T3 to connect to the server instance listening at localhost:7001.
	The utility has been updated to use HTTP or HTTPS if you specify them. Note that you must enable HTTP tunneling if you want to use these protocols. See "Configuring the HTTP Protocol" in the *Administration Console Help*.

Cluster

Change Request Number	Description
CR086319	The Node Manager was writing an unnecessary stack trace on shutdown of a Managed Server running under an Administration Server under certain conditions.
	The problem occurred because - `HttpUrlConnection.getInputStream` doesn't return because the Managed Server waits for a certain period of time before returning the response to a query. This was intentional behavior to provide almost continuous monitoring of the Managed Server. When the Managed Server shut down, however, the delayed servlet response was not sent and the socket was closed. The Node Manager subsequently tried to re-open the connection but failed to connect in `openServer` call because the server was not there. In response to this, a `ConnectException` was thrown and logged by `openServer`.
	The Node Manager no longer writes the unnecessary stack trace.
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.
	<pre><Dec 26, 2002 7:17:55 PM EST> <Error> <HTTP Session> <Error looking up session for id:2LcR6Ool2IC5qNtFZhThJClYVYEXt9KWG2ciXmAXZ4XiBAHPiHx4!-137950 51 94!stcasfi01b.usa-ed.net!8001!7002!-1127797657!stcasfi02b.usa-e d. 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...</pre>
	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.

Change Request Number	Description
CR094837	The following sequence was causing a `ClassCastException`: define a server, assign that server to a cluster, create a Unix machine with the same name as the server, assign them to each other, stop and restart the Administration Server, and select the server name from the cluster or from the Servers list. Trying to bring up the Managed Server also caused a `ClassCastException`.
	Fixing a problem in a process that resolves an UnresolvedMBean during startup solved this problem.
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.
CR105482	Starting a new Managed Server in a cluster that contained two running Managed Servers caused a `ConcurrentModificationException` on one of the running servers.
	A code change fixed the problem by making the members of a cluster meet synchronization requirements.
CR105698	If two clusters have between them two distributed queues with the same JNDI name, and a message-driven bean is deployed on both the clusters, messages were being consumed in only one of the clusters.
	A message-driven bean now goes through the entire collection of distributed queues in a domain and adds their members appropriately.

Change Request Number	Description
CR106384	An Administration Server and two Managed Servers were in a cluster. Application `application1.ear` contained EJBs and Web applications. The EJBs were deployed to the cluster. The Web applications were deployed to a VirtualHost, which was in turn targeted to the cluster.
	`application1.ear` contained `application1.war`, which was configured as DefaultWebApp of the VirtualHost. In addition to this, each cluster node had its own DefaultWebApp.
	Deployment succeeded, but attempts to undeploy `application1.ear` resulted in the exception below, and the application could not be redeployed, even if the entire domain was restarted.
	`Exception caught for task Unprepare application application1 on mycluster,application1: Start server side stack trace: java.lang.reflect.UndeclaredThrowableException: java.lang.reflect.InvocationTargetException: javax.management.ListenerNotFoundException: listener: ServletContext(id=559103557,name=DefaultWebApp,context-path=)`
	The Web container was trying to undeploy its DefaultWebApp with the undeployment of every WAR. The first WAR succeeded and the second WAR failed because the DefaultWebApp is already undeployed.
	The code was changed so that if no context for the targeted Web application exists on the http server, the default Web application is not undeployed. This resolved the problem.
CR107273	In a cluster, if you set the value '0' for `MulticastTTL`, WebLogic Server ignored the setting and took the default value as 1, throwing an error like the following:
	`"clusterdomain:Name=mycluster,Type=Cluster" is smaller than the minimum allowed: 1`
	A code change resolved the problem by allowing the value to be '0'.

Change Request Number	Description

CR134233, CR102655	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.

CR110385	To prevent an incorrect port number from being constructed in the `HomeHandleImpl` class while constructing the URL in NetworkChannel between two Managed Servers, `listenPort` and `sslListenPort` are now initialized with default values.

Connector

Change Request Number	Description
CR069426	A configuration exception was thrown when deploying a RAR if max-capacity and initial-capacity are equal. The check for capacity-increment in .xpi for weblogic-ra.xml was altered so that its value is not considered when max-capacity and initial-capacity are equal. Also, the capacity increment can now be set to zero if max-capacity and initial-capacity are equal.
CR086862	Unexplainable log messages appeared in the Connector log file, for example: "Unable to locate context: java:/comp/env/WebLogic Server-connector-resref". The Connector team eliminated voluminous logging messages as an enhancement.

Change Request Number	Description
CR091106	The Weblogic Server embedded LDAP server stopped replicating data to Managed Servers after redeploying the connector RAR file.
	A code change causes improved replication behavior.
CR100707	In versions of WebLogic Server prior to 7.0 SP3, using JCE threw a `java.lang.ClassCastException` with `java.naming.CompositeName` was thrown. When using JCA in WebLogic Server 8.1, resource-ref entries (Auth, SharingScope) returned `null` for Web applications.
	This problem occurred because of an internal error, and was resolved with a code fix.
CR101761	When using JCA in versions of WebLogic Server prior to 7.0 SP3, a `java.lang.ClassCastException` with `java.naming.CompositeName` was thrown. When using JCA in WebLogic Server 8.1, resource-ref entries (Auth, SharingScope) returned `null` for Web applications.
	This problem occurred because of an internal error, and was resolved with a code fix.

Change Request Number	Description
CR101906	A client invoked a method on an EJB. When the connection failed between the ManagedConnection (MC) and the EIS while in XA transaction state, the container called XAResource.rollback and ManagedConnection.cleanup every 60 seconds after the transaction had already been rolled back. Subsequent calls made to the EIS resulted in a java.lang.NullPointerException exception:

```
<Mar 21, 2003 5:50:52 PM PST> <Error> <Connector> <190041> << JCA
Resource Adapter for ClearPath MCP_eis/comsRAJNDINAMEBrazil > The
returned ManagedConnection instance is null> <Mar 21, 2003 5:50:52
PM PST> <Info> <EJB> <010051> <EJB Exception during invocation
from home:
drummer.CedarBank.CedarBankBeanBrazil_ris6bf_HomeImpl@12fc36
threw exception: java.lang.NullPointerException
java.lang.NullPointerException at
drummer.CedarBank.CedarBankBeanBrazil.cedarBank(CedarBankBeanBra
zil.java:130) at
drummer.CedarBank.CedarBankBeanBrazil_ris6bf_EOImpl.cedarBank(Ce
darBankBeanBrazil_ris6bf_EOImpl.java:46) at
drummer.CedarBank.CedarBankBeanBrazil_ris6bf_EOImpl_WebLogic
Serverkel.invoke(Unknown Source) at
weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.java:
362) at
weblogic.rmi.cluster.ReplicaAwareServerRef.invoke(ReplicaAwareSe
rverRef.java:114) at
weblogic.rmi.internal.BasicServerRef$1.run(BasicServerRef.java:3
13) at
weblogic.security.service.SecurityServiceManager.runAs(SecurityS
erviceManager.java:785) at
weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServerRe
f.java:308) at
weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecuteRe
quest.java:30) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)
```

WebLogic Server had failed to wait for the transaction to complete before trying to destroy the connection.

A code fix resolved the problem.

Change Request Number	Description
CR106388	During deployment of a RAR that uses the `ra-link-ref` element, called a 'logical' RAR, the logical RAR deployment was not getting the correct pool name or max-capacity as specified in its `weblogic-ra.xml` deployment descriptor. This caused unexpected behavior in WebLogic Integration.
	The element now functions properly.

Core Server

Change Request Number	Description
CR070887	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 WebLogic Server's `HttpURLConnection` redirect logic to ensure process redirects in accordance with the setting of `InstanceFollowRedirects`.
CR071415	When the security manager in WebLogic Server was enabled and a policy file was set, the path /usr/lib was being prepended to javac during JSP compilation, resulting in a java.security.AccessControlException.
	`Executable.resolveExecutable()` was searching the wrong path when trying to determine the location of the `javac` compiler.
	The code was changed so that `Executable.resolveExecutable()` searches the absolute path specified in JavaCompiler before searching `java.library.path`.

Change Request Number	Description
CR077170	In WebLogic Server 6.1 SP03, stopping a Managed Server after its Administration Server was shut down caused an exception. The problem occurred with this sequence of actions:
	1. Start an Administration Server and a Managed Server.
	2. Shut down the Administration Server.
	3. Invoke the 'stop' method on the ServerConfigMBean of the Managed Server.
	This exception resulted:
| | ```
Unexpected Exception Start server side stack trace:
java.rmi.ConnectException: Unable to get direct or routed
connection to:
'-19546889165621794S:dwarkamai:[7001,7001,-1,-1,7001,-1,-1]:OAMd
omain:adminServer' at
weblogic.rmi.internal.BasicOutboundRequest.sendReceive(BasicOutb
oundRequest.java:109) at
weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.java:
127) at...
``` |
	The shutdown command from weblogic.Admin was successful.
	The problem was resolved by catching the exception that occurs when shutting down a Managed Server after the Administration Server is down.
CR083209	Provided a code fix to allow the NTSockedMuxer to support Windows XP.

Change Request Number	Description
CR083762	Failures occurred due to `NoSuchObjectExceptions` when attempting to get a connection from a data source on the client. This message was thrown:

	`Error dropping table: Unexpected Exception java.rmi.NoSuch`
	`ObjectException: The object identified by: '275' could  not be`
	`found. Either it was has not been exported or it has been collected`
	`by the distributed garbage collector. at`
	`weblogic.rjvm.BasicOutboundRequest.sendReceive(BasicOutboundRequ`
	`est.java:109) at weblogic.rmi.internal.BasicRemoteRef.`
	`invoke(BasicRemoteRef.java:120)at weblogic.jdbc.common.`
	`internal.RmiDataSource_WebLogic Servertub.getConnection(Unknown`
	`Source)`
	`at com.bea.cts.CTSDeployment.executeSQL(CTSDeployment.`
	`java:2445)at com.bea.cts.CTSDeployment.undeploy(CTS`
	`Deployment.java:711) at com.sun.cts.harness.Suite`
	`Synchronizer.continueToUndeployApps(SuiteSynchronizer.java:1082)`
	`at com.sun.cts.harness.SuiteSynchronizer.`
	`undeployApps(SuiteSynchronizer.java:982) at com.sun.cts.`
	`harness.SuiteSynchronizer.doDeployment(SuiteSynchronizer.java:28`
	`9) at com.sun.cts.harness.CTSGUIHarnessObserver.`
	`starting(CTSGUIHarnessObserver.java:167) at javasoft.sqe.`
	`javatest.TestRunner$Notifier.starting(TestRunner.java:287) at`
	`javasoft.sqe.javatest.TestRunner$Worker.runTest`
	`(TestRunner.java:239)at javasoft.sqe.javatest.TestRunner$`
	`Worker.run(TestRunner.java:221)`

	Analysis revealed that the data source was garbage collected, although the client still had a reference to it.
	The problem was solved by a code fix to garbage collection, to ensure that objects that clients refer to are not garbage collected.
CR087364	Changes to the Enable Tunneling attribute in the WebLogic console require a server restart for the new value to take effect.
	The problems was solved by a code fix that allows the user to enable tunneling without restarting the server.

Change Request Number	Description
CR087808, CR095487	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`, and a second server instance on 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(BasicOutb
oundRequest.java:85) at
weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwareRe
moteRef.java:262) at
weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwareRe
moteRef.java:229) at
weblogic.rmi.internal.ProxyStub.invoke(ProxyStub.java:35) at
$Proxy45.lookup(Unknown Source) at
weblogic.jndi.internal.WLContextImpl.lookup(WLContextImpl.java:3
41) 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(ServletS
tubImpl.java:265) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:200) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(Web
AppServletContext.java:2546) at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequ
estImpl.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.

A code change prevents RJVM shutdown when detecting duplicate connections, and `CMD_REQUEST_CLOSE` does not get delivered to server.

Change Request Number	Description
CR088022	The WebLogic Server Administration Console does not provide connection information when starting the Administration Server and Managed Servers via the Node Manager.
	A code fix allows the user view the connection information by right clicking the server name in the navigation tree and selecting View Connections.
CR089454	Users can now throttle incoming traffic by limiting the request queue waiting for execute queue, with the -D flag.
CR092454	Methods that access COMMO beans via the `CommoMBeanServerProxy` class from a client virtual machine sometimes triggered a Null Pointer Exception when attempting to ascertain whether an `ObjectName` was an MBean instance or an MBean Type object name, after the code that set the value for "bean" had already been bypassed by `Kernel.isServer()`.
	A code change prevents the premature bypass.
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. The following 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:1049)
at
weblogic.security.SSL.SSLSocket.sendAlert(SSLSocket.java:1007)
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 ensures 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. |

Change Request Number	Description
CR093320	An internal problem with the servlet container caused a source to be returned even if the class could not be loaded. This was discovered while testing a WebLogic Server 7.0SP2 client which got a database connection from a JDBC connection pool on a WebLogic Server 8.1 server. The error prevented the client from getting the database connection.
	The problem was resolved with a code fix.
CR094697	A code fix resolved a problem in which the server was shut down properly while the client sees an error message "Server XYZ failed to shutdown successfully ...".
CR094724	WebLogic Server native libraries on HPUX were being compiled using `cfront,` instead of `aCC`. `aCC` is the ANSI C compiler which replaced `cfront` in 1998.
	As a result incompatible runtime libraries were loaded (`libC.2` from `cfront` compilation, and `libCsup.2` from SUN's Java, compiled with `aCC`). Incompatible runtime libraries can cause crashes.
	The problem was solved by changing WebLogic Server builds to use `aCC` for hpux11.
CR095487	A t3 client attempting multiple `getInitialContext()` calls with different server URLs that refer to the same server. In a normal situation, the client determines that it is a duplicate connection, sends a CLOSE message to the server and shuts its own end of the connection. The server will remove this duplicate connection. In some situations, the CLOSE message gets queued because of an asynchronous IDENTIFY request and the client closes its end of the connection without waiting for the CLOSE message to be delivered. The server detects that this socket has been removed and issues a shutdown on the ConnectionManager which shuts down the otherwise healthy RJVM.
	A code fix prevents RJVM shutdown when closing duplicate connections when a CLOSE message does not get delivered.

Change Request Number	Description
CR096049	A system with an Administration Server and two Managed Servers in a cluster has trouble restarting a Managed Server. After rebooting the machine on which one of the Managed Servers is running, the Managed Server is fails to restart and the following exception is thrown:

```
<Jan 22, 2003 5:27:49 PM EST> <Error> <Deployer> <149204> <The
Deployment framework was unable to register with the Data
Replication Service. RegisterException due to underlying
exception java.rmi.RemoteException: Failed to send message to URL
t3://admtest1:7001; nested exception is:
javax.management.InstanceNotFoundException:
myDomain:Location=AdminServer,Name=***servername***,ServerRuntim
e=AdminServer,Type=DeploymentTaskRuntime
```

This problem was solved by a code fix to allow the Managed Server to load applications based on the configuration MBeans.

Change Request Number	Description
CR096291	A startup class that calls one EJB business method on `doSomethingOnA` of BeanA. This method gets the home interface for BeanB and this leads to the following ClassCastException:

```
<29 janv. 03 13:21:19 CET> <Info> <EJB> <010051> <EJB Exception
during invocation from home:
test_ccex.a.BeanABean_124zg1_HomeImpl@5c5ca2 threw exception:
java.lang.ClassCastException: Cannot narrow remote object to
test_ccex.b.BeanBHome java.lang.ClassCastException: Cannot
narrow remote object to est_ccex.b.BeanBHo me at
weblogic.iiop.PortableRemoteObjectDelegateImpl.narrow(Portabl
eRemoteObjectDelegateImpl.java:223) at
javax.rmi.PortableRemoteObject.narrow(PortableRemoteObject.ja
va:132) at
test_ccex.a.BeanABean.doSomethingOnA(BeanABean.java:49) at
test_ccex.a.BeanABean_124zg1_EOImpl.doSomethingOnA(BeanABean_
124zg1_EOImpl.java:46) at
java.lang.reflect.Method.invoke(Native Method) at
com.csg.am.dcf.weblogic.startup.BeanExecutor.startup(BeanExec
utor.java:183) at
weblogic.t3.srvr.StartupClassRunner.invokeStartup(StartupClas
sRunner.java:141) at
weblogic.t3.srvr.StartupClassRunner.invokeClass(StartupClassR
unner.java:122) at
weblogic.t3.srvr.StartupClassRunner.access$0(StartupClassRunn
er.java:113) at
weblogic.t3.srvr.StartupClassRunner$1.run(StartupClassRunner.
java:86) at
weblogic.security.service.SecurityServiceManager.runAs(Securi
tyServiceManager.java:744) at
weblogic.t3.srvr.StartupClassRunner.run(StartupClassRunner.ja
va:71) at java.lang.Thread.run(Thread.java:479)
```

A code fix resolved a classloading issue in `RemoteObjectReplacer`.

| CR097077 | When using a custom realm that made many outbound RMI calls, the reader threads became blocked by making outbound calls, leaving no threads for reading the response. This happens from the BootServicesImpl.invoke method from within the RJVM layer on a reader thread.

`BootServicesImpl` was moved into the RMI layer so that it can dispatch to the default execute queue. |

Change Request Number	Description
CR099307, CR102154 CR100469, CR104186, CR104352, CR103539	When closing duplicate t3 connections, there was a chance that the threads (the PosixSocketMuxer and RJVM ConnectionManager) could enter a deadlock state because of the order is which locks are acquired. The FDRecord lock is no longer held during dispatching the requests in the muxer.
CR099489	A `java.lang.ClassCastException` was thrown when the home interface was narrowed using PortableRemoteObject after looking it up from the InitialContext. A code fix correctly narrowed the returned types.
CR100177	The use of `weblogic.security.service.ServerResource` was inconsistent with the way server resources were created in the Administration Console and with the documentation. When used to secure a T3 server, the name of the WebLogic resource was always "T3Srvr." In all other cases, the name of the server being acted upon was used as the name of the WebLogic resource. This problem was resolved by always using the name of the server being acted upon as the name of the WebLogic resource.
CR101720	With connection filtering enabled, `Connection rejected` messages accumulated and filled up the log file. The new ConnectionLogger property lets users configure whether to log such messages to the log file.
CR102848	The CoreHealthMonitor was holding the ExecuteThreadManager lock while performing significant work, resulting in deadlocks and Administration Console freezes. The code was changed so that CoreHealthMontor now uses ExecuteThreadRuntimeMBean to get a list of stuck threads and avoids logging from all places in the ExecuteThreadManager if ETM lock is held.

Change Request Number	Description
CR103721	A t3 client would time out while getting an `InitialContext` from a clustered EJB via a content switch, and the following exception was thrown:
	``` javax.naming.CommunicationException. Root exception is java.net.ConnectException: t3://10.161.39.60:9001: Bootstrap to: 10.161.39.60/10.161.39.60:9001' over: 't3' got an error or timed out at weblogic.rjvm.RJVMFinder.findOrCreate(RJVMFinder.java:180) at weblogic.rjvm.ServerURL.findOrCreateRJVM(ServerURL.java:262) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext(WLInitialContextFactoryDelegate.java:323) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext(WLInitialContextFactoryDelegate.java:221) at weblogic.jndi.WLInitialContextFactory.getInitialContext(WLInitia lContextFactory.java:149) at javax.naming.spi.NamingManager.getInitialContext(NamingManager.j ava:668) at javax.naming.InitialContext.getDefaultInitCtx(InitialContext.jav a:246) at javax.naming.InitialContext.init(InitialContext.java:222) at javax.naming.InitialContext.<init>(InitialContext.java:198) at Test.main(Test.java:34) ```
	Including support for t3 clients with load balancers resolved this problem.
CR103967	Child threads of WebLogic Server execute threads were not inheriting the correct context classloader. This problem was common when a servlet or an EJB created a timer (`java.util.Timer`).
	WebLogic Server execute threads maintain their own context classloader and child thread of these execute threads will not inherit since `java.lang.Thread` directly assigns the member variable of its own to the child threads.
	Setting the same context classloader to `java.lang.Thread` resolved this problem.

Deployment

Change Request Number	Description
CR089760	The `weblogic.Deployer` utility generated "alternate descriptor" errors when deploying an EAR file. For example: `Module:` *module_name* `Error: Alternative descriptor` *descriptor_name* `can't be found.` This problem occurred because WebLogic Server did not look for the descriptor files in the directory where the EAR was extracted. `weblogic.Deployer` was fixed to correctly deploy EARs.
CR090678	All Managed Servers in a domain automatically redeployed EJBs when you restarted the domain's Administration Server. A code fix ensures that restarting an Administration Server does not redeploy EJBs on the domain's Managed Servers.

CR090678	WebLogic Server required an incorrect `CODEBASE` value to access applets deployed as part of an EAR file. For example, an applet packaged inside a Web Application, MyWar.war, and deployed as part of an EAR file, `MyEar.ear`, required a `CODEBASE` similar to:
	
	`CODEBASE=/bea_WebLogic Server_internal/classes/MyEar@MyWar.war`
	
	Where `MyWar.war` is the name of the WAR file that contains the applet. This was required even if the Web Application explicitly defined a `CONTEXT-ROOT` of `MyWar`. This problem occurred because WebLogic Server deployed using the URI of the Web Application, rather than its `CONTEXT-ROOT`.
	
	The code was fixed to deploy Web Applications using the `CONTEXT-ROOT`, if one is specified the EAR file's `application.xml` descriptor. For example, if the above EAR file specified a `CONTEXT-ROOT` of `MyWar` for the Web Application, the correct `CODEBASE` for the applet would be:
	
	`CODEBASE=/bea_WebLogic Server_internal/classes/MyEar@MyWar`

CR100857	Refreshing a Web application with `weblogic.Deployer` (with `-activate -upload` flags) was not working properly. The web app was deactivated, rolled back and destroyed before being loaded. This was a regression introduced by another fix. The code was changed to roll back the Web application under the appropriate circumstances.

Change Request Number	Description
CR101755	When the Web container is Tomcat 4.0 and the EJB container is WebLogic Server7.0SP1, and two Web applications which have different names and use different classloaders but the same classes are deployed on Tomcat, the first Web application is able to invoke the EJB, but the second Web application fails to invoke it, with a ClassCastException.
	A new -D flag enforces clients to generate stubs on the context classloader. The -D flag is as follows:
	`-Dweblogic.LoadStubUsingContextClassLoader="true"`
CR101760	Deleting a Web application from a Managed Server that had been stopped resulted in errors similar to the following:
	`<Warning> <Management> <149311> <Rejecting deployment operations to non-running server, mgd1>`
	`  <Warning> <Deployer> <149004> <Failures detected initiating weblogic.management.ManagementException: Rejecting deployment operations to non-running server, mgd1 task for application Remove application MyWebApp on mgd1>`
	The Web application was not deleted.
	A code change fixed the problem by eliminating dangling `WebAppComponentMBean` references from the `WebServer` and `VirtualHost MBeans`.
CR102928	Web applications in an EAR deployed to a VirtualHost did not redeploy on domain restart.
	A code change has resolved the problem so that applications that target a Virtual Host restart when the domain restarts.
CR108934	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp.

Domain Configuration Wizard

Change Request Number	Description
CR101796	In WebLogic Server 7.0 SP2, Configuration Wizard templates used in Silent Mode created unusable domains.
	A code change to `silent.xml` fixed the problem.

EJB

Change Request Number	Description
CR056023	CMP entity EJBs did not time out when corresponding rows in the database were locked with a `select for update` or `update` statement. The code was changed so that a cancel is sent to any running JDBC Statements before rolling back the transaction. The EJBs now time out when the transaction timeout happens even if the corresponding rows in the database are locked.

CR063837	The following error was thrown when the EJB Container tried to verify the existence of database tables and columns that did not exist:
	
	`weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[`
	`Table: Cmp_Birthday Full Table Check failed, but table all columns`
	`were found! ] at`
	`weblogic.ejb20.utils.TableVerifier.verifyTableAndColumnsExist(`
	`TableVe rifier.java:371) at`
	`weblogic.ejb20.utils.TableVerifier.verifyTableExistsAndCreate`
	`Maybe(Ta bleVerifier.java:391)`
	
	The problem was solved with a code fix.

CR068572	EJB Query Language (EJB-QL) now supports case-insensitive searching with the new Upper and Lower functions.
CR081512 CR097906 CR094280	A `Finder` method did not find updates made inside a transaction when the concurrency strategy was Optimistic; that is, the `include-updates` functionality was failing. The code was changed so that `include-updates` works with an Optimistic concurrency strategy, for Oracle databases.

Change Request Number	Description
CR081951	A client called a stateless session bean, which in turn called a bean-managed persistence entity bean. The entity bean returned a TreeMap to the session bean. This provoked the following exception under load conditions:
	`Tue Jul 16 11:46:54 PDT 2002:<E> <Adapter> Exception thrown by rmi server: [386851657836047808S172.17.24.65:[8001,8001,8002,8002,8001,-1]/2 62] java.util.ConcurrentModificationException at java.util.TreeMap$Iterator.next(TreeMap.java:1023) at java.util.TreeMap.writeObject(TreeMap.java:1498) at java.lang.reflect.Method.invoke(Native Method) at java.io.ObjectOutputStream.invokeObjectWriter(ObjectOutputStream .java:1864)...`
	Analysis revealed that the entity bean began processing a subsequent call before the current call to TreeMap's `WriteObject()` method had completed serialization of the object and returned, resulting in the `ConcurrentModificationException`. This behavior resulted from the bean being locked at the EJB container layer and serialized at the RMI layer.
	The problem was resolved by using RMI activation callbacks to return EJBs back to the pool.
CR087151	A stateless session bean in a cluster with `stateless-bean-is-clusterable` set to `False` generated errors such as the following:
	`<Mar 14, 2003 3:35:00 PM PST> <Error> <Cluster> <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 deployed from one server.>`
	The JNDI bindings are no longer replicated for the non-clustered stubs.

Change Request Number	Description

CR087151	For a stateless session bean, when the `weblogic-ejb-jar.xml` contained:
	
	`<stateless-bean-is-clusterable>False</stateless-bean-is-clusterable>`
	
	and the bean was deployed on a cluster, this error was generated:
	
	`<Mar 14, 2003 3:35:00 PM PST> <Error> <Cluster> <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 deployed from one server.>`
	
	A code fix ensures that JNDI bindings for non-clustered stubs are not replicated.

CR088156, CR124471	When the `PersistenceManagerImpl.releaseResources()` method was closed, JDBC statements whose connection was being used in another thread—`weblogic.transaction.internal.ServerTransactionImpl.commit()`—to commit the transaction, the following deadlock resulted: `FOUND A JAVA LEVEL DEADLOCK: --------------------------- "ExecuteThread: '180' for queue: 'default'": waiting to lock monitor 0xcbb28 (object 0xdee1d070, a oracle.jdbc.driver.OraclePreparedStatement), which is locked by "ExecuteThread: '73' for queue: 'default'" "ExecuteThread: '73' for queue: 'default'": waiting to lock monitor 0xcbc78 (object 0xdec416b8, a oracle.jdbc.driver.OracleConnection), which is locked by "ExecuteThread: '180' for queue: 'default'"`  Research revealed that the EJB container was closing statements prematurely. The code was changed so that `PersistenceManagerImpl.releaseResources()` closes the statements at the appropriate time.
CR089759	When calling `context.getCallerPrincipal()` from `ejbStore()` using `UserTransaction` context, the following exception was 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(AccountBean.java:99)` This problem was a result of changes introduced in SP02 to comply with section 21.2.5.1 of the EJB specification. The problem was corrected.

Change Request Number	Description

CR089822	The `-classpath` option to `weblogic.ejbc`, which sets a CLASSPATH used during compilation and overrides the system or shell CLASSPATH, now works. Previously, it was failing with the following errors:
	
	`"ERROR: Error from ejbc: Unable to load a class specified in your ejb-jar.xml: examples.ejb20.basic.statelessSession.TraderBean"`
	
	`"ERROR: ejbc found errors"`

CR089936	WebLogic Server called `ejbload()` incorrectly for a clustered container-managed persistence bean deployed to a WebLogic Server cluster; `ejbLoad()` was called for the bean each time the bean was requested, after the EJB had been modified from another server in the same cluster. The concurrency strategy was `Optimistic` and both `cache-between-transactions` and `home-is-clusterable` were enabled. This problem occurred because the bean's state was not correctly updated after the first call to `ejbLoad()`. The code was fixed to correctly update the bean state.
CR090104	An EJB specified an isolation level in its descriptor, and got an exception at runtime. WebLogic Server was checking for the wrong vendor workaround; it was checking for `supportsTxIsolation` instead of `supportsTxIsolationUponEnlistment`. In addition, different database vendors and drivers have different rules and behaviors when (re)setting the isolation level while the transaction is active. For example, when using the WebLogic Server jDriver/XA for Oracle, the isolation level can only be set when creating the transaction branch, but not when joining or resuming the branch. Code changes rectified both problems.

Change Request Number	Description
CR090192	When a stateless session bean with `max-beans-free-pool` set to `No LIMIT` was copied to the `applications` directory during deployment, the deployment failed with the following error: `<Nov 5, 2002 4:59:00 PM EST> <Error> <Deployer> <BEA-149201> <The Slave Deployer failed to complete the deployment task with id 7 for the application _appsdir_BasicStatelessTraderBean_jar. weblogic.management.ApplicationException: Prepare failed. Task Id = 7 { Module Name: BasicStatelessTraderBean, Error: [EJB:011025]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 S erver examples to ensure the format is correct. The error was: Path .weblogic-ejb-jar.weblogic-enterprise-bean.caching-descriptor.ma x-beans-in- free-pool.: [EJB:010136]Param must be an integer.`  The problem was with parsing the white space in "`NO LIMIT`."  The code was fixed so that if `max-beans-in-free-pool` is specified as `NO LIMIT`, it is now set to a maximum value of int.
CR090515	The Administration Console incorrectly reported, in the Waiter Total Count field, the number of clients waiting for access to entity beans with exclusive concurrency and stateful session beans. This count is now correctly decremented when a client acquires the lock or times out.
CR091263	A stateless session bean attempted to update related beans. After the transaction had been rolled back due to an Oracle constraint violation SQL Exception, an attempt to access the stateless session bean triggered this error: `com.cpships.ecomm.exception.FatalException: EJB Exception: javax.ejb.TransactionRolledbackLocalException: Illegal Reentrant call to RelatedCompanyHomeRemote with primary key: 30: weblogic.ejb20.InternalException: Illegal Reentrant call to RelatedCompanyHomeRemote with primary key: 30` Research determined that after the `weblogic.ejb20.manager.DBManager.remove()` method was executed, the `setBusy` flag, which determines whether or not the bean can be used (that is, whether or not the bean is busy), was not correctly reset to `False`. This problem was fixed.

Change Request Number	Description
CR091352	`ejbc` code generation failed for four-level relationship caching, with the following exception:

	``
	Found 4 semantic errors compiling
	"D:/labs/cluster70/src/com/bea/ps/cmr_orders/ejbcgen/com/bea/ps/
	cmr_orders/CustomerBean_1q1rpd__WebLogic_CMP_RDBMS.java": 789.
	if (__WL_bean__orderLines != null) <-------------------> ***
	Error: No entity named "__WL_bean__orderLines" was found in this
	environment. 791.
	__WL_bean__orderLines.__WL_add__WL_item_field_(__WL_eo);
	<-------------------> *** Error: "__WL_bean__orderLines" is
	either a misplaced package name or a non-existent entity. 1310.
	if (__WL_bean__orderLines != null) { <-------------------> ***
	Error: No entity named "__WL_bean__orderLines" was found in this
	environment. 1312.
	__WL_bean__orderLines.__WL_add__WL_item_field_(__WL_eo);
	<-------------------> *** Error: "__WL_bean__orderLines" is
	either a misplaced package name or a non-existent entity.
	``
	
	Analysis revealed that `RDBMSCodeGenerator.java` trimmed an extra character when creating the `prevCmrFieldName` string, if caching was more than three levels.
	
	A fix to `RDBMSCodeGenerator.java` solved the problem.

CR091436	When message-driven beans using external `InitialContextFactory()` calls for lookups made repeated connection attempts and the JMS provider was inaccessible, connections and other resources accumulated rapidly, draining server resources. Further research revealed that `JMSConnectionPoller()` was not properly closing its `InitialContexts`. This problem was fixed in the code.
CR091774	When a container-managed relationship `Finder` method was executed, WebLogic Server incorrectly attempted to load the state from the database for second accesses to same the entity EJBs, even though the EJBs had `cache-between-transactions` set to `True` and were using Optimistic concurrency. This problem was fixed in the code.
CR094073	BMP entity bean handling was changed to conform to the EJB 2.0 specification. Per the EJB specification, the EJB container now synchronizes the entity bean state—by invoking EJBStore—before running a Finder method. This does not affect the findByPrimaryKey method.

Change Request Number	Description
CR094524	In WebLogic Server 6.1 SP3, read-only EJBs with a many-to-one container-managed relationship caused a `LockTimedOutException`. The problem was resolved by modifying the `if condition` check from `rdb.isReadOnly()` to READONLY_EXCLUSIVE_CONCURRENCY check.
CR094629, CR098386	Two beans with a container-managed relationship of one-to-one were created in the same transaction. A ghost SQL `SELECT statement` was issued, even though the `delay-database-insert-until` was set to `Commit` for both beans. In high-volume situations, the true benefit of batch insert was not realized because of the extra `SELECT`, resulting in inferior performance. The code was changed so that the extra `SELECT` is no longer issued.
CR094776	If a create was attempted for an entity EJB but then failed, the Pool Beans in Use Count reported in the Administration Console was incremented as if the create operation had succeeded. This problem has been fixed.

Change Request Number	Description
CR094861	If an EJB contained more than one attribute with a `String[]`, the `ejbc` compiler would throw the following error:
| | ```
C:\public\383031>java weblogic.ejbc
ejb11_basic_containerManaged.jar
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:499: byteArray is already defined in
ejbFindByPrimaryKey(java.lang.Str ing) byte[] byteArray =
__WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:611: byteArray is already defined in
ejbFindAccount(double) byte[] byteArray = __WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:732: byteArray is already defined in
ejbFindBigAccounts(double) byte[] byteArray =
__WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:838: byteArray is already defined in
ejbFindNullAccounts() byte[] byteArray = __WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:1236: byteArray is already defined in
ejbLoad() byte[] byteArray = __WL_rs.getBytes(5); ^ 5 errors Exec
failed .. exiting
``` |
	This problem occurred only with EJB 1.1.
	The code was changed so that unique variables are created for byte arrays in the generated code.
CR095023	EJB deployment descriptors can be bound and referenced using the token "`${APPNAME}`" in the JNDI name. But the `LocalJNDI` Name did not support the "`${APPNAME}`" token to be resolved at deployment time. Now `LocalJNDI` name is transformed for the application name token replacement during deployment.

Change Request Number	Description

CR095030	Attempts to deploy a stateful session EJB in WebLogic Server 7.0 SP1 failed with the following `ClassCastException`:
	
	`Unable to deploy EJB: MetaDataBS from`
	`csam_dcf3_server_ejb_depl.jar: java.lang.ClassCastException:`
	`weblogic.ejb20.deployer.SessionBeanInfoImpl at`
	`weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.setMethodDescri`
	`ptors(ClientDrivenBeanInfoImpl.java:745) at`
	`weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.prepare(Client`
	`DrivenBeanInfoImpl.java:907) at`
	`weblogic.ejb20.deployer.EJBDeployer.setupBeanInfo(EJBDeployer.ja`
	`va:1041) at`
	`weblogic.ejb20.deployer.EJBDeployer.prepare(EJBDeployer.java:124`
	`5) at`
	`weblogic.ejb20.deployer.EJBModule.prepare(EJBModule.java:242) at`
	`weblogic.j2ee.J2EEApplicationContainer.prepareModule(J2EEApplica`
	`tionContainer.java:1504) at`
	`weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationCo`
	`ntainer.java:690) at`
	`weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationCo`
	`ntainer.java:576) at`
	`weblogic.management.deploy.slave.SlaveDeployer.processPrepareTas`
	`k(SlaveDeployer.java:1064) at`
	`weblogic.management.deploy.slave.SlaveDeployer.prepareUpdate(Sla`
	`veDeployer.java:732) at`
	`weblogic.drs.internal.SlaveCallbackHandler$1.execute(SlaveCallba`
	`ckHandler.java:24) at`
	`weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at`
	`weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)`
	
	The problem occurred because `setMethodDescriptors()` was incorrectly identifying session beans as entity beans.
	
	The problem has been fixed.

CR095090	If, within the same transaction, `delay-database-until-insert` was set to `Commit` and `ejbRemove()` was called, the attempt to remove the bean would fail with a `java.rmi.NoSuchObjectException`. The code was changed to check the container cache to remove the bean when `delay-database-insert-until` is set to `Commit`.

Change Request Number	Description
CR095141	In a clustered configuration and with an Not Recently Used (NRU) caching strategy, the container was not passivating a stateful session bean correctly when the number of bean instances instantiated exceeded `max-beans-in-cache`.

In a clustered environment, WebLogic Server uses a ReplicatedStatefulSessionManager to create a new NRUCache.

Research revealed that the Replicated Stateful SessionManager did not register itself with the cache and therefore beans were not moved from the active to the inactive and then to the free queue correctly, resulting in CacheFullExceptions.

A code fix resolved the problem. |
| CR095173 | The `idle-timeout-seconds` element determined 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 this element to determine how long to wait before removing passivated EJBs from the disk. However, some users wanted stateful session beans to remain on disk longer than `idle-timeout-seconds`. They wanted to specify how long stateful session beans stay idle in the cache and how long they stay idle on disk using two different elements.

WebLogic Server 7.0 Service Pack 3 introduces the new element `session-timeout-seconds`, which specifies how long the EJB container waits before removing an idle stateful session bean from disk. |
| CR095290 | An application's attempt to update—inside a transaction—an entity EJB with `delay-database-insert-until` set to `Commit` and a concurrency strategy of Optimistic failed with a `NullPointerException`.

The problem was that with a `delay-database-insert-until` value of `Commit`, WebLogic Server did not initialize the Optimistic concurrency version column at the time it created the EJB.

The code was changed so that the Optimistic concurrency version column is initialized for bulk inserts. |

Change Request Number	Description
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,ServerRu ntime=jpserver,Type=EJBMessageDrivenRuntime] at weblogic.management.runtime.RuntimeMBeanDelegate.register(Runtim eMBeanDelegate.java:96) at weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM BeanDelegate.java:83) at weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM BeanDelegate.java:53) at weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM BeanDelegate.java:63) at weblogic.ejb20.deployer.MessageDrivenRuntimeMBean.<init>(Message DrivenRuntimeMBean.java:18) at weblogic.ejb20.deployer.MessageDrivenBeanInfoImpl.deploy(Message DrivenBeanInfoImpl.java:450) at weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.java: 1299) at weblogic.ejb20.deployer.Deployer.deploy(Deployer.java:1005) ``` The problem resulted from non-unique `MessageDrivenRuntimeMBean` names. The code was changed to create unique names.
CR095632	A plain Java client created an entity EJB, read the EJB using a Finder method and read-only beans, updated the entity bean, and then read entity EJB again, using a stateless session bean as a facade. When the `transaction-attribute` was set to `Supports`, the client read updated values. However, when `transaction-attribute` was set to `Required` or `RequiresNew`, the client read stale values. Research revealed that because the cache was set to a timeout of 0,the Finder method was not refreshing the values of timed out beans. The code was changed so that it refreshes the beans even when the cache timeout is set to 0.
CR096182	Dynamic EJB-QL incorrectly passed long arguments as `MAX_INT` to the underlying SQL, resulting in an unsuccessful search. A code fix resolved the problem.

Change Request Number	Description
CR096395	The container was failing to call `unsetEntityContext` when the entity bean was undeployed. An undeploy method was added to the `BaseEntityManager` to clean up the pool when an entity bean is undeployed. This resolved the problem.
CR096800	When WebLogic Server 6.1 throws an EJB exception to an WebLogic Server 7.0 invoker, the invoker receives a serialization error instead accompanied by scary error messages. This is because WebLogic Server 6.1 shipped with the wrong serial version unique identifier (SVUID) for EJBException. WebLogic Server 7.0 now identifies when it is communicating with WebLogic Server 6.x and accepts and sends the correct SVUID over such connections. The earlier resolution attempt that shipped with 7.0 was flawed.
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.
CR097890	WebLogic Server scheduled multiple `JMSPoller` triggers at the same time; they all dominated the default JMS queue, provoking a deadlock. The code was changed to prevent scheduling duplicate triggers.
CR097913	Combining COUNT and EXISTS in a dynamic EJB-QL request resulted in invalid SQL and this exception: `[java] javax.ejb.FinderException: Exception in dynamicQuery while preparing or executing statement: 'weblogic.jdbc.rmi.SerialStatement@62b30' [java] java.sql.SQLException: ORA-00937: not a single-group group function [java] [java] java.sql.SQLException: ORA-00937: not a single-group group function` An unnecessary column was being added to the main select list while parsing a subquery. A fix to the parser code solved the problem.

Change Request Number	Description
CR098188	A message-driven bean consumed messages from a JMS queue on another server. That JMS server died. Inside of `weblogic.ejb20.internal.MDListener`, the `UserTransaction.commit method` threw an exception. `MDListener` continued running. Subsequent invocations of the MDB from JMS failed in `UserTransaction.begin` because there was already a transaction on the thread. The code was changed so that `MDListener` suspends the current transaction before starting a new one.
CR098923	The `mdbPoolInfo.start()` method was changed so that it is run on every message-driven EJB in a JAR, even if the start failed for the first MDB in the JAR.
CR099420	The `ejbc` compiler failed with the following error when a two-dimensional array was specified as an argument: `ERROR: Error from ejbc: 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 `makeMethodParameter()` method of the `weblogic.ejb20.deployer.mbimpl.MethodDescriptorImpl` class was modified to generate the appropriate method parameter signature depending on the dimension of the array passed.
CR100822	The behavior of clustered message-driven beans has changed to optimize performance. Previously, only standard cluster load-balancing algorithms were used when an MDB connected to JMS, so that JMS connections were balanced across the cluster. When possible, an MDB steers its JMS connection toward the same JVM that is hosting the messages from which the bean receives connections. This reduces the number of hops made by the MDB when it is accessing the resource.
CR100832	Column version numbers were not updating correctly with Optimistic concurrency enabled, because blind writes were not checked . The code was changed so that blind writes are checked with Optimistic concurrency.

Change Request Number	Description

CR101226	Automatic table creation was failing intermittently with the following error in WebLogic Server 7.0 SP1:
	
	`weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[`
	`Table: Cmp_Birthday Full Table Check failed, but table all columns`
	`were found! ] at`
	`weblogic.ejb20.utils.TableVerifier.verifyTableAndColumnsExist(Ta`
	`bleVe rifier.java:371) at`
	`weblogic.ejb20.utils.TableVerifier.verifyTableExistsAndCreateMay`
	`be(Ta bleVerifier.java:391) at`
	`weblogic.ejb20.cmp.rdbms.RDBMSPersistenceManager.populateFieldSQ`
	`LType Map(RDBMSPersistenceManager.java:499) at`
	`weblogic.ejb20.cmp.rdbms.RDBMSPersistenceManager.setup(RDBMSPers`
	`isten ceManager.java:122) at`
	`weblogic.ejb20.manager.BaseEntityManager.setupPM(BaseEntityManag`
	`er.ja va:192) at`
	`weblogic.ejb20.manager.BaseEntityManager.setup(BaseEntityManager`
	`.java :168) at`
	`weblogic.ejb20.manager.DBManager.setup(DBManager.java:123) at`
	`weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.deploy(ClientDr`
	`ivenB eanInfoImpl.java:807) at`
	`weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.java:`
	`1234)`
	
	Research revealed that `preparedStatement` was used to fire the query. Because the prepared statement can get deleted from the Oracle database, intermittent assertion errors can occur.
	
	The code was revised to use `Statement` instead of `preparedStatement`.

CR101248	CMP beans leaked JDBC connection pools. The problem occurred when `ejbFindByPrimaryKey()` called `releaseResources()`. A change was made to the `RDBMSCodeGenerator` so that, if `ResumeTransaction` fails, WebLogic Server releases resources to avoid a potential JDBC connection leak.
CR102180	The `ejbc` compiler was failing a JAR file that contained one hundred EJBs. The problem was fixed by using the `@tempfile` feature, which is based on the number of files passed independent of the operating system.

Change Request Number	Description
CR102481	In some bean-managed stateful session beans, an `IllegalStateException` occurred with bean-demarcated transactions when the number of bean instances is greater than `max-beans-in-cache` when the cache is full and activations and passivations occur. A code change results in the creation of one replacer per passivation request, so that each thread has its own replacer.
CR104046	With `delay-database-insert-until` set to `Commit`, the Optimistic concurrency version column was not initialized when a "multi-table" bean was created. The code was changed so that `RDBMSCodeGenerator.perhapsAssignOptimisticField()` now returns generated code that sets the version column for all the tables that the CMP bean is mapped to.
CR105857	EJBs with Bean-Managed Persistence and with concurrency set to exclusive called ejbFindByPrimaryKey rather than finding the bean in the cache. A code change resolved the problem so that the EJB finds the bean in the cache.

Interoperability

Change Request Number	Description
CR095804	When a WebLogic Server 7.0 SP2 client attempted to get the Coordinator remote object from a WebLogic 8.1 server instance, a `ClassCastException` was thrown while casting the remote stub to the CoordinatorOneway interface. This prevented commit handoff to the 8.1 coordinating server. A code change that altered classloading functionality fixed this problem.

JDBC

Change Request Number	Description
CR087870	In previous releases of WebLogic Server 7.0, JDBC MultiPools with the High Availability algorithm did not fail over when the DBMS for the first connection pool was removed from the network. A code fix solved the problem.
CR088525	In previous WebLogic Server 7.0 Service Packs, JSP parameters with a value of "/////" were interpreted by the `getParameter()` as having the value "/". The problem was resolved with a code change.
CR090379	In previous releases of WebLogic Server 7.0, erroneously creating multiple ResultSets that point to a single JDBC object through a JDBC connection pool resulted in an `OutOfMemoryError`. Each ResultSet resulted in an RMI object. Not all RMI objects were closed when the ResultSet was closed. 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 was traced to 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.

Change Request Number	Description
CR092791	In WebLogic Server 6.1, it was not possible to use specific Oracle objects (Array, Struct, and others) through a connection pool based on the Oracle Thin Client driver. The objects returned by the Oracle Thin Client driver are not serializable nor remote and therefore cannot be passed over RMI.
	In WebLogic Server 7.0 SP3, a new package, `weblogic.jdbc.vendor.oracle`, contains proxies for these objects and allows them to be passed through the connection pool.
CR093038	Starting with WebLogic Server 7.0 SP3, WebLogic Server provides support for Oracle Virtual Private Databases (VPDs). A VPD is an aggregation of server-enforced, application-defined fine-grained access control, combined with a secure application context in the Oracle 9i database server.
	For more information, see "Programming with Oracle Virtual Private Databases" in Programming WebLogic JDBC.
CR093649	The `ejb2jsp` utility was generating incorrect tag classes for methods that contained parameter types other than String.
	The problem was fixed by changing the tag library generator to use `getAttribute` rather than `getAttributeString`.
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 of at the specified starting index.
	A code fix solved the problem.
CR094729	The `JDBCConnectionPoolRuntimeMBean.getStatementProfiles()` method did not filter results by poolName for the MBean instance it was called against. If you called the `getStatementProfiles` method, the results included statements for all connection pools for which tracing had been activated. This was incorrect because the `JDBCConnectionPoolRuntimeMBean` instance is specific to a single connection pool.
	This problem was resolved by modifying the `JDBCConnectionPoolRuntimeMBean.getStatementProfiles()` method to correctly filter results by poolName for the MBean instance it is called against.
CR095059	Rather than returning the actual time a JDBC operation took to execute, `JDBCStatementProfile.getTimeTaken()` always returned zero, and the start and end time of the operation were always equal.
	A code fix to `weblogic/jdbc/common/internal/ProfileStorage.java` solved the problem.

Change Request Number	Description
CR095962	Threads trying to replace their JDBC connection were unnecessarily synchronized in the `resetThisOne()` method of `ResourceAllocator`, resulting in a waste of resources. The `ResourceAllocator` code was changed so that threads replace their connections in parallel.
CR095994	Previous versions of WebLogic Server lacked connection leak profiling and connection leak detection logic in `weblogic.jdbc.rmi.internal.ConnectionImpl`. Code changes to `SerialConnection.java` and `ConnectionImpl.java` solved this problem.
CR096268	When the DBMS was not available, `makeConnection() in ConnectionEnvFactory.java` was trying to connect DBMS twice on behalf of the client. This has been fixed so that only one connection attempt is made.
CR096922	Under load conditions, when WebLogic Server was calling `ResourceAllocator.markBorrowed()` and JMX was calling `ConnectionPoolRuntimeMBeanImpl.getConnectionDelayTime()`, a deadlock condition resulted. A code fix to `ResourceAllocator.java` solved the problem.
CR097832	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 was turned on, with the following stack trace: `"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(ConnectionEnv.java:571) at weblogic.common.internal.ResourceAllocator` Both the `weblogic.jdbc.common.internal.ConnectionEnv.destroy` and `weblogic.common.internal.ResourceAllocator.release` are synchronized methods. The problem was solved with a code fix to `weblogic/jdbc/common/internal/ConnectionEnv.java`.

Change Request Number	Description
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 were used, all the other connections in the pool were reserved for testing at the expiration of the refresh test minutes. At that point, if the 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 only reserves and tests one connection at a time, and releases 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, the pool avoids testing pool connections that are known to have successfully connected to the DBMS within that period. This accelerates `refresh` and `testConnsOnReserve`.
CR099507	While closing prepared statement, the `cleanUpStatementForReUse()` method of `ConnectionEnv.java` called `clearBatch()`, which is not implemented by MS SQL Driver. This resulted in the following exception:
	`java.sql.SQLException: This JDBC 2.0 method is not implemented exception.`
	The code was changed so that the harmless call is permitted and no exception is thrown.
CR101419	In `build-jdbc.xml`, `RmiDataSource` was not built as clusterable. This prevented `DataSource` failover.
	Building `DataSources` and `TXDataSources` as clusterable solved the problem.

Change Request Number	Description
CR102698	Attempts to create a connection pool with `weblogic.Admin CREATE_POOL` resulted in the following exception: ``` ./wlg-create-pool.sh No permission to create ConnectionPool Start server side stack trace: weblogic.common.ResourceException: No permission to create ConnectionPool at weblogic.jdbc.common.internal.JDBCService.createPool(Ljava.util. Properties;Lweblogic.secur ity.acl.internal.AuthenticatedSubject;)V(Unknown Source) at weblogic.jdbc.common.internal.ConnectionPool.createPool(Ljava.ut il.Properties;Lweblogic.se curity.acl.internal.AuthenticatedSubject;)V(Unknown Source) at weblogic.jdbc.common.internal.ConnectionPool.createPool(Ljava.ut il.Properties;)V(Unknown S ource) at weblogic.jdbc.common.internal.ConnectionPool_WebLogic Serverkel.invoke(ILweblogic.rmi.spi.InboundReque st;Lweblogic.rmi.spi.OutboundResponse;Ljava.lang.Object;)Lweblog ic.rmi.spi.OutboundResponse;(Unknown Source) at weblogic.rmi.internal.BasicServerRef.invoke(Lweblogic.rmi.intern al.MethodDescriptor;Lweblo gic.rmi.spi.InboundRequest;Lweblogic.rmi.spi.OutboundResponse;)V (Unknown Source) at weblogic.rmi.internal.BasicServerRef$1.run()Ljava.lang.Object;(U nknown Source) at weblogic.security.service.SecurityServiceManager.runAs(Lweblogic .security.acl.internal.Aut henticatedSubject;Lweblogic.security.acl.internal.AuthenticatedS ubject;Ljava.security.PrivilegedExcep tionAction;)Ljava.lang.Object;(Unknown Source) at weblogic.rmi.internal.BasicServerRef.handleRequest(Lweblogic.rmi .spi.InboundRequest;)V(Unk nown Source) at weblogic.rmi.internal.BasicExecuteRequest.execute(Lweblogic.kern el.ExecuteThread;)V(Unknown Source) at weblogic.kernel.ExecuteThread.execute(Lweblogic.kernel.ExecuteRe quest;)V(Unknown Source) at weblogic.kernel.ExecuteThread.run()V(Unknown Source) at java.lang.Thread.startThreadFromVM(Ljava.lang.Thread;)V(Unknown Source) ``` Research revealed that WebLogic Server was requiring `aclName`—which is deprecated—as an attribute. A code fix resolved the problem.

Change Request Number	Description
CR102740	The `XAPreparedStatementCacheSize` in the Administration console for JDBC connection pools is now exposed.
CR102792	An SP2 change allowed the isolation level for drivers that do not allow isolation-level setting. Because the DB2 JDBC driver allows changes to the isolation level the check is not called and thus a change of isolation level is done at an inappropriate time.
	This means that EJBs that set the `isolation-level` in deployment descriptors can throw an exception, even if the isolation level in the deployment descriptor is the same as the default 'READ-COMMITTED'.
	A code change fixed this by preventing the isolation level from being set if it is the same as the current level.
CR103256	A code change resulted in performance improvement in JDBC regarding bubblecaches in `JDBCSessionContext` and `JDBCSessionData`.
CR103321	The `getConnectionsTotalCount` method from `JDBCConnectionPoolRuntimeMBean` did not work as expected. It should return the total number of JDBC connections in this `JDBCConnectionPoolRuntimeMBean` since the pool was instantiated, but instead was returning the maximum number of connections at any point in time, equal to or less than the `MaxCapacity`.
	Code changes to `getTotalConnections()` in `ResourceAllocator.java` fixed the problem.
CR103560	`PreparedStatement` for SQL UPDATE did not work after using `CallableStatement` on the same connection. `getUpdateCount` returned zero when nothing has changed on the database. This happened when setting the `weblogic.oci.min_bind_size` property for the JDBC connection. A code change resolved the problem.
CR104523	`DatabaseMetaData.getDriverVersion()` returned an outdated version string. A code fix resolved the problem.
CR106522	The WebLogic Server jDriver for MS SQLServer can mistakenly accept the `jdbc:microsoft:...` URL for the Microsoft JDBC driver. If a JVM tries to use both drivers, loading the WebLogic Server driver first will prevent the Microsoft driver from being used.
	This jDriver is has been deprecated.
CR106767	When applications try to close JDBC objects more than once, WebLogic Server now throws an exception.

jDriver

Change Request Number	Description
CR088387	Using a XADataSource on top of jDriver resulted in shrinking heap size until OutOfMemoryError.
	In the weblogic.jdbc.oci.Connection class, LOB fields of result sets in a transaction are registered under a connection through the Connection.addLob() method. The registered LOBs are freed (along with the corresponding object in native jDriver library) when one of these conditions occurs:
	• Connection.commit()/Connection.rollback()/Connection.close() is called at the connection level (this Connection is weblogic.jdbc.oci.Connection).
	• An SQL statement is executed and the connection is set to autoCommit mode (i.e., non-TX Data Source or a direct connection from the pool).
	• The return code from an OCI SQL call indicates that a commit/rollback at the database level has occurred.
	When an XADataSource is used, none of the above conditions apply. As a result, LOB fields of jDriver in result sets were never released in the JVM heap.
	This problem was corrected by a code change in weblogic.jdbc.oci.xa.DataSrcThreadInfo.java to call closeLob ().
CR090025	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.<init>(Connection.java:246):
	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.
	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.
	This problem was corrected by a code fix to weblogic/db/oci/OciConnection.
CR090761	In previous releases of WebLogic Server 7.0, a StringIndexOutOfBoundsException occurred in the WebLogic jDriver for Oracle. The problem occurred occasionally in multiple queries with medium-sized (hundreds of rows) ResultSets. A code fix solved the problem.

Change Request Number	Description
CR091151	In WebLogic Server 6.1 SP03 and SP04, jDriver for Oracle `ResultSet.getBigDecimal()` method did not return correct value. For example, the value of `9999999999.999999` was rounded to `9999999999.999998`. This problem was solved with a code fix.

CR096730	The WebLogic jDriver for MS SQL Server that shipped with WebLogic Server 7.0., 7.0 SP1, and 7.0 SP2 threw the following error when used to connect to a MS SQL Server 2000 SP3 database:
	
	 `java.sql.SQLException: I/O exception while talking to the server, java.io.EOFException Unable to connect, please check your server's version`
	 `and availability.`
	
	 This problem was solved with a code fix.

CR098071	The version of the Oracle Thin driver 9.2.0 that was 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.

JMS

Change Request Number	Description
CR101298	Long-lived JMS connections lacked a periodic heartbeat check. Following a code change, when JMS is idle the connection pings the database every five minutes to keep connection fresh.
CR080289	In WebLogic Server 7.0, the Messaging Bridge did not work properly with a null `UserPassword`. When a messaging bridge was configured with a non-null `UserName` and null `UserPassword`, it failed to start with no clear error messages. This problem was resolved with a code fix.

Change Request Number	Description

CR083933	In WebLogic Server 6.1 SP03, a JMS server threw a `NullPointerException` under heavy load conditions:
	
	`<Warning> <JTA> <XA resource [JMS_JMSServer1JDBCStore] has not responded in the last 120 second(s).>`
	`<Warning> <JTA> <Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 >`
	
	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 and a Sybase driver was used for the connection pool for the JMS JDBC store.
	
	The problem is resolved by a code fix.

CR089583	In WebLogic Server 6.1 SP02, a JMS exception occurred when attempting to unsubscribe a transacted durable subscriber: `weblogic.jms.common.JMSException: 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)` Analysis revealed that WebLogic Server was incrementing the unacknowledged message count twice for every message and decrementing it only once. This always left the unacknowledged `messagecount > 0` and thus the exception when trying to unsubscribe the durable subscriber. This problem was solved with a code fix to unacknowledged message counts and pending durable subscribers.

Change Request Number	Description
CR089682	In WebLogic Server 6.1 SP03, the Messaging Bridge connection retry values (minimum, maximum, and increment) did not work correctly. The problem occurred with a Messaging Bridge configured to send messages from one WebLogic JMS server to another, with both JMS servers running on the same WebLogic Server instance. Untargeting one of the JMS servers resulted in this error: \<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 ```<Error> <Connector> <Error granting connection request.>\n<Error> <Connector> <Error granting connection request.>\n<Error> <Connector> <Error granting connection request.>\n<Error> <Connector> <Error granting connection request.>\n<Error> <Connector> <Error granting connection request.>``` After one of the default parameters was changed, the increments were correct. However, the max value was never reached. With the default values, the maximum reached was 55. A code fix corrected the logic for the retry values.
CR090403	In WebLogic Server 7.0 SP02, when enqueuing multiple messages in a JMS queue within a JTA transaction, if the transaction was actually rolled back after the enqueuing, then the Administration Console displayed incorrect information. A code fix adjusted the `receivedCounts` on transaction rollbacks.
CR091195	In WebLogic Server 6.1 SP04 *only*, 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 the server was shut down and restarted. This problem was solved with a code fix to MDB acknowledgments when the EJB transaction descriptor is set to "NotSupported".

Change Request Number	Description
CR091827	In WebLogic Server 6.1 SP02, when a connection pool was created using the Oracle Thin Driver, and a JMS JDBC Store was configured to use this pool (with JDBC logging enabled), produced a `Connection has already been closed` SQL exception. After this error message, nothing was logged in the JDBC log file. A code fix prevents the JDBC logging errors.
CR091844	In WebLogic Server 7.0 SP01, calling a "stop" on a `QueueConnection` on which there was a `ServerSessionPool` that was still receiving messages from a JMS queue, caused an endless loop and CPU utilization of 100%. The problem was resolved by a fix to the code.
CR092110	In WebLogic Server 7.0 SP01, a JMS connection factory's Flow Control settings were configured to set the `FlowMinimum` to 2 and the `FlowMaximum` to 8. After restarting the server, the following configuration exception was thrown: ```<Critical> <WebLogic Server> <000364> <Server failed during initialization. Exception:weblogic.management.configuration.ConfigurationExcepti on: invalid value '8' for attribute 'FlowMaximum' of MBean "mydomain:Name=MyJMS Connection Factory,Type=JMSConnectionFactory".``` The problem was resolved by removing the legal checks on FlowMinimum and FlowMaximum fields. This is a limitation in the legal checking that can be done in this release.
CR092464	In WebLogic Server 6.0 SP02 and SP04, when a durable topic subscriber was connected to a JMS server and a message was 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` was 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 was 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.

Change Request Number	Description
CR092468	In WebLogic Server 7.0 SP01, analysis of the JMS connection factory's `FlowMinimum` minimum allowable value of 0 (for Flow Control) was shown to have negative impact on message producers. Although a value of 0 seemed to indicate that a producer could be completely stopped from sending messages, this was not the case as messages would keep trickling through. The problem was resolved by changing the minimum allowable value for FlowMinimum to 1.
CR093177	In WebLogic Server 7.0 SP01, the Messaging Bridge Administration Console labels, online help, and log messages conflicted with how the Maximum Idle Time and Incremental Delay attributes were measured. These attributes are measured in "seconds", not "milliseconds". The problem was resolved by correcting the Administration Console labels, online help, and log messages to reflect that these attributes are measured in "seconds".
CR093712	In WebLogic Server SP04, after a messaging bridge's Maximum Idle Time setting was reached, messages were logged into the server name log file. If there were numerous messaging bridges configured, then the log file would fill up quickly and possibly grow quite large. These messages are now only generated when run-time debugging is enabled for the messaging bridge.
CR094414	In WebLogic Server 7.0 SP02, although the Messaging Bridge documentation stated that a value of -1 for the Messaging Bridge's Execute Thread Pool Size field was valid and that it should be used to disable this thread pool to force all messaging bridges to use the WebLogic Server default thread pool, a value of -1 for this field was not actually allowed. A code fix allows the Execute Thread Pool Size to be set to "-1".

CR094670	In WebLogic Server 7.0 SP01, the following security error message was generated when using the Messaging Bridge to interoperate between WebLogic Server domains that did not establish a trusted security relationship:
	`java.lang.SecurityException: Invalid Subject:`
	`principals=[searchuser]`
	WebLogic Server security now allows synchronous operations between untrusted domains.

CR096229	In WebLogic Server 7.0, enforced security on physical destinations was not effective when accessed through distributed destinations. For example, if a distributed destination had two physical queue members, SecQ1 and SecQ2, and a user was not given JMS security privilege to "send" messages to either physical queue, the user was still able to send messages successfully through the distributed destination. The problem was solved by checking security against the physical destination members of the distributed destination.

Change Request Number	Description
CR096854	A user set up a messaging bridge on one server in the cluster and then migrated the bridge to another server in the cluster. Starting up the second server triggered the following exception:
	`(java.lang.Exception: java.lang.NullPointerException at weblogic.jms.bridge.internal.MessagingBridge.getConnections(MessagingBridge.java:714) at weblogic.jms.bridge.internal.MessagingBridge.execute(MessagingBridge.java:919) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)`
	The messaging bridge skipped one of the initialization steps when it was targeted to a migratable target; therefore, the bridge failed to get the resource adapter information from the configuration. The NullPointerException occurred when the bridge later tried to access a resource adapter.
	The problem was resolved by fixing the messaging bridge initialization code for migratable targets.
CR097451	In WebLogic Server 7.0 SP01, a stateless EJB's topic messages were delivered to a `TopicSubscriber` client before a connection was started.
	A code fix prevents messages from being delivered before a connection is started.
CR097538	In WebLogic Server 7.0 SP01, the following parsing selector error was produced when the length of the JMS selector attribute was less than 3:
	`weblogic.jms.common.InvalidSelectorException: Error parsing selector`
	The problem was resolved by a code fix.
CR097963	In WebLogic Server 7.0, when a JMS server had an inbound and an outbound queue, and 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 all done within a transaction initiated by the MDB.
	The end of the transaction occurred when the MDB enqueued a message on the outbound queue. If the JMS Server was stopped and restarted, messages were lost. Over a load of 10,000 messages, some messages (a small quantity) were lost.
	A code fix forces the XASession to perform a CLIENT_ACK acknowledgement.
CR098049	In WebLogic Server 7.0 SP01, duplicate message IDs could be generated by JMS.
	The problem was resolved by a code fix.

Change Request Number	Description
CR098268	In WebLogic Server 7.0 SP01, a message-driven bean would open a XAConnection if it was using an XAConnectionFactory. This meant that for a non-transactional MDB, the container connected using an XASession, and did not realize it was client-acknowledge, so it did not call "acknowledge" and messages did not get acknowledged.

A code fix changed this behavior so that regardless of the type of connection factory returned, the MDB container only creates an XAConnection and a XASession if XA is required. |

Change Request Number	Description
CR098280	The session creation method behavior has been changed when using a `XAQueueConnection` or `XATopicConnection` object to create a non-XA Session. Prior to this change, the behavior was as follows: • `XAQueueConnection.createQueueSession` creates a `XAQueueSession` • `XATopicConnection.createTopicSession` creates a `XATopicSession` In both cases, the user-specified acknowledge mode and transacted flag were ignored and replaced with the `AUTO_ACKNOWLEDGE` mode and a transacted setting of *false*. Whereas, the new behavior causes the `XAQueueConnection.createQueueSession` and `XATopicConnection.createTopicSession` methods to behave *exactly* the same as the corresponding methods on `QueueConnection` and `TopicConnection`, as follows: • `QueueConnection.createQueueSession` creates a `QueueSession` • `TopicConnection.createTopicSession` creates a `TopicSession` • `XAQueueConnection.createQueueSession` creates a `QueueSession` • `XATopicConnection.createTopicSession` creates a `TopicSession` Furthermore, the user-specified acknowledge mode and transacted flag settings are honored for each of these four methods. The four connection methods listed above behave differently if the `XAServerEnabled` flag is enabled on the connection factory. If this flag is enabled, then all four methods create an `XAQueueSession` (or `XATopicSession`) session if invoked on the server, and a non-XA `QueueSession` or `TopicSession` session if invoked on the client. The resulting session honors the user-specified acknowledge mode but ignores the transacted flag because the resulting session supports XA. Note that in previous versions of WebLogic Server 7.0, connection objects that were created from a connection factory with the `XAConnectionFactoryEnabled` flag enabled behaved as if they were `XAQueueConnection` or `XATopicConnection` objects. With the new behavioral change, this behavior is now invisible unless you explicitly cast the connection factory to `XAQueueConnectionFactory` or `XATopicConnectionFactory` and called one of the `createXA` methods. Prior to this change, if you set the `XAConnectionFactoryEnabled` flag on your connection factory, you would have noticed different behavior from the `createQueueSession` and `createTopicSession`, even if you did not cast the connection factory to one of the XA connection factory classes.

Change Request Number	Description
CR098280	In WebLogic Server 7.0 SP01, on the client and the server side, calling the `createSession()` method on a XAConnection object resulted in a XASession object.
	The behavior was changed so that calling the `createSession()` on an XASession object results in a non-XA "Session" object.
CR098692	In WebLogic Server 6.1 SP04, the synchronous receiving of messages on a JMS queue stopped after a garbage collection/transaction timeout.
	The problem was resolved by refactoring the JMS dispatcher code to improve thread management.
CR099455	In WebLogic Server 7.0 SP01, when a message that was being pushed to a consumer arrived at the front-end session, but was destined for a consumer that had already been closed, the messages could have been lost because the list was being inadvertently truncated.
	The problem was resolved by a code fix to prevent a potential race condition.
CR100015	`dispatch-policy` is now honored for message-driven beans. This means message-driven beans can now be assigned to a user-defined execute queue.
CR100076	In WebLogic Server 7.0, when a JMS server was migrated from a WebLogic Server instance (server1) to another instance (server2), the JMS server was deactivated on server1 and all its destinations were suspended. However, the in-memory messages lists were not cleaned up when the destinations were suspended. As a result, when the JMS server was migrated back to server1, all the messages that existed on the JMS server before the migration would show up even if they were dequeued from server2. Once the in-memory messages cleaned up on server1 after the JMS server is migrated to server2, there should no be any duplicate messages.
	The problem was resolved by cleaning up the in-memory messages lists when a JMS server is migrated to another WebLogic Server instance.
CR100083	In WebLogic Server 7.0, the run-time MBeans associated with a JMS server (`JMSServerRuntimeMBean` and `JMSDestinationRuntimeMBean`) were not unregistered with the Administration Server after the JMS server was suspended. As a result, if the JMS server was targeted to a migratable target, two instances of those MBeans would show up in the Administration Console monitoring pages after the JMS server was migrated from one server to another. One instance was on the original server instance, and another was on the server where the JMS server was migrated.
	The unregistration/registration process for run-time MBeans is now handled correctly.

Change Request Number	Description
CR101298	If a long-idle JMS JDBC connection was marked as "bad" (when for example, a firewall's time-to-live expires), JMS attempted to use the bad connection, and failed. The failure would either be immediate or would take upwards of 8-10 minutes to return, depending on tuning parameters.

This problem was resolved by modifying the code so that if JMS is idle, the database is pinged every five minutes to maintain the connection. Five minutes was selected because it is expected to be shorter than any firewall timeout. |
| CR103213 | 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.

The failure to enlist the transaction is now communicated back to client, which will be able to report that the commit failed. |
| CR105237 | When using JMS with the IIOP protocol, abnormal disconnection by a client was not being properly detected in the server. This causes problems with stateful information associated with the client, like the JMS client identifier.

Abnormal disconnection is now detected and handled appropriately. |

Change Request Number	Description
CR105337	Under load conditions, a test JMS application was encountering the following exception:
	java.lang.OutOfMemoryError weblogic,jms.common.TransactionRolledBackException: at weblogic.jms.backend.BEConsumer.expireTimeout(BEConsumer,java:1620) at weblogic.jms.backend.BEXATranEntryBlockingConsumer.startRollback(BEXATranEntryBlockingConsumer.java:72) at weblogic.jms.backend.BEXAResource.rollback(BEXAResource,java:1205) at weblogic.transaction.internal.ServerResourceInfo.rollback(ServerResourceInfo.java:1400) at weblogic.transaction.internal.ServerResourceInfo.rollback(ServerResourceInfo.java:664) at weblogic.transaction.internal.ServerSCInfo.startRollback(ServerSCInfo.java:365) at weblogic.transaction.internal.ServerTransactionImpl.localRollback(ServerTransactionImpl.java:1521) at weblogic.transaction.internal.ServerTransactionImpl.globalRollback(ServerTransactionImpl.java:2142) at weblogic.transaction.internal.TransactionImpl$1.execute(TransactionImpl.java:1656) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:211) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:198) weblogic,jms.common.JMSException: Only one thread may use a JMS Session at a time. at weblogic.jms.frontend.FESession.rollbackAfterRecover(FESession.java:1773) at weblogic.jms.frontend.FESession.recover(FESession.java:1373) at weblogic.jms.frontend.FESession.invoke(FESession.java:2253) at weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Request.java:609) at weblogic.jms.dispatcher.DispatcherImpl.dispatchSyncNoTran(DispatcherImpl.java:339) at weblogic.jms.client.JMSSession.recoverGuts(JMSSession.java:868) at weblogic.jms.client.JMSSession.rollback(JMSSession.java:632) at com.ncr.crm.framework,jms.DefaultJmsQueueReceiver.rollback(DefaultJmsQueueReceiver.java:184) at com.ncr.crm.framework,jms.DemoMessageConsumer.run(JmsMemoryLeakDemo.java:163) at java.lang.Thread.run(Thread.java:479) ----------- Linked Exception ----------- weblogic,jms.common.JMSException: Only one thread may use a JMS Session at a time. at weblogic.jms.frontend.FESession.recover(FESession.java:1219) at weblogic.jms.frontend.FESession.invoke(FESession.java:2253) at weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Request.java:609) at weblogic.jms.dispatcher.DispatcherImpl.dispatchSyncNoTran(DispatcherImpl.java:339) at weblogic.jms.client.JMSSession.recoverGuts(JMSSession.java:868) at weblogic.jms.client.JMSSession.rollback(JMSSession.java:632) at com.ncr.crm.framework,jms.DefaultJmsQueueReceiver.rollback(DefaultJmsQueueReceiver.java:184) at...
	Research revealed that the WebLogic JMS garbage collection functionality was slow to reclaim memory when dealing with long linked lists; the code was changed to break up the lists. This resolved the problem.

Change Request Number	Description
CR105980	A memory leak occurred when creating a `QueueReceiver` for each message on a Distributed Queue and persisted even after the `QueueReceiver` had been shut down. A code change corrected memory leaks in `BasicServiceOffer` and `DistributedDestinationImpl` objects.
CR106789	A memory leak occurred because unused JMS objects remained open. The problem was resolved by the following fixes: -Normalized the destination for all the Producers created on the same destination. -Removed the listener from the correct `DispatcherWrapperState` when substituting the `DispatcherWrapperState` for FEProducer -Removed the listener from the correct `DispatcherWrapperState` when closing the `FEProducer` unconditionally -Removed the leak for Temporary Destinations on `FrontEnd` when the `BackEnd` goes down.

JNDI

Change Request Number	Description
CR084381	JNDI Context 's list("") method did not return the correct class name. WebLogic Server now returns the actual class name when using the "list" method.
CR087237	RMI failover took a long time when kill -STOP was invoked on the server. A code change reduced the time needed for failover.
CR093700	JNDI tree was not updated properly on rebind. JNDI unbind of Clusterable RemoteObjects from one of the cluster nodes caused the objects to still be accessible when the calls landed on the node from which the object was unbound from. This is now fixed and a next available node will be picked to make the remote method invocations.

CR093799	When a Context.lookup on a composite name was done, a NameNotFoundException was obtained if the name was not resolved, either fully or partially.
	When RemainingName is a CompositeName, it should have an enumeration. The remaining name was treated as a single string and WebLogic Server passed '.' as a separator when constructing the CompositeName. Now WebLogic Server properly replaces any '.' separators with '/' to ensure the enumeration is properly created.
CR103148, CR110890	In WebLogic Server 7.0 SP2 a startup class failed to access the local instance `MBeanHome` with `LOCAL_JNDI_NAME` (`weblogic.management.home.localhome`) with `LoadBeforeAppDeployments` set to true. It failed with the following error message:
	`Unable to resolve 'weblogic.management.home.localhome' Resolved:` `'weblogic.management' Unresolved:'home'`
	Moving the bindings of the other JNDI names into the AdminService so that they are available to startup classes fixed the problem.
CR106853	Attempting to bind a Dynamic Proxy object into the local JNDI tree caused a ClassNotFoundException.
	A code change fixed the problem by implementing `resolveProxyClass()` and using a thread context classloader to load interfaces from the application.

JSPs and Servlets

Change Request Number	Description
CR083597	When the content type was set using the page directive, such as `<%@ page contentType="text/html;charset=UTF-8" %>` then the content type was not set properly and the desired output was not obtained.
	This problem was eliminated by a code change.
CR088044	When "extended format" is selected for the HTTP logging option and the server is restarted, a Null Pointer Exception was thrown.
	Adding code to open the access.log file during logManagerHttp's creation time eliminated this problem.

CR088350	`HttpProxyServlet` was unable to reach third-party server.
	The problem occurred because the server does not send the ContentLength and keeps sending data forever. If there is no content length and it is not chunk transferred, the server reads until the end of stream.
	HttpProxyServlet has been changed to read and write byte by byte.
CR088747	The user-agent was unable to re-establish a WebLogic Server session following a request to a Web application that does not explicitly set CookiePath in `weblogic.xml`.
	WebLogic Server uses the first cookie with a name equal to Cookie Name specified in weblogic.xml, even if a second cookie exists from the same domain and specifies a path that matches the request URL.
	In the past, WebLogic Server searched the cookie vector backwards for the JSESSIONID; this search has been changed to run forward.
CR088759	The `redirect-with-absolute-url` feature did not work for `j_security_check`. The Location header in a response was still the absolute path.
	The code has been altered so that `redirect-with-absolute-url` will be honored in FormSecurityModule.
CR089347	For Network Channels, the `getServerPort` method returns the value of the `FrontendHTTPPort` if the `FrontendHTTPPort` value is set in the `WebServer` element in `config.xml`.
	When the HOST header contains a port other than the WebLogic Server default listen port (for example when a proxy sits in front of WebLogic Server), the `FrontendHTTPPort` can be specified to allow `getServerPort()` to return the WebLogic Server port instead of the one specified in the HOST header. However, this FrontendHTTPPort value was returned for all getServerPort() calls, including those for Network Channels which are running on a different port.
	The code has been altered to allow the use of call `request.getAttribute("weblogic.servlet.network_channel.port")` to return the value of the port on which the request was received.
CR089582	An incorrect Date header was returned if there was no Host header in the request.
	This has been corrected by setting the request invoke time before error messages are sent.
CR089868	Applying the patch created to fix CR083654 made multi-bytes headers inaccurate.
	This has been fixed by adding the `UseHeaderEncoding` option to WebServer.
CR090188	For chunked transfer, WebLogic Server was treating the chunk size prepended to data.
	`setChunking` has been moved out of `mergePostParams` of `ServletRequestImpl` to `MuxableSocketHttp` right after `request.setInputStream`.

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	If the Http Header did not have X_WEBLOGIC_CLUSTER_LIST, the ServerList was not created for the code that checks for X_WEBLOGIC_CLUSTER_HASH.
	This condition was fixed by enabling the server to store the hash if the list is sent after hash in the response headers.
CR091759	`RequestDispatcher.forward` dropped version numbers in the URL header.
	This was fixed by setting `processPathParameters` to false in `setRequestURI()` when doing `forward()`.
CR091831	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.
	Turning off keep-alive on the server made the problem disappear. Increasing Keep-Alive periods still showed the problem with different length of sleep.
	Checks were added to ensure that the HttpClient is non-null before updating the timestamp.
CR091922	The servlet container did not pass the login exception to the error page.
	Now the error JSP can access the exceptions.
CR092039	WebLogic Server threw `UnsupportedEncodingException` for the extra quoted value in charset with JSP tag.
	This problem was resolved by adding support for HTTP spec for header attribute values.

CR092039	In WebLogic Server 6.1 SP03, the `JspParser` threw an `UnsupportedEncodingException` for the extra quoted value in charset with JSP tag:
	
	`<%@ page contentType="text/html; charset=\"Shift_JIS\"" %>`
	
	resulted in this error:
	
	`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`
	
	WebLogic Server did not correctly support the HTTP specification.
	
	The problem was resolved by adding support for quoted strings and comments to `Content-Type` header charset attribute value.

CR092173	File servlet `GET` functionality now supports byte ranges.
CR092255	WebLogic Server stripped the quotation marks from cookies. This made the cookie values unreadable under certain circumstances. The code has been changed to allow quotation marks to remain in cookies.
CR092377	When setting: <user-data-constraint> <description>USE SSL</description> <transport-guarantee>CONFIDENTIAL</transport-guarantee> </user-data-constraint> in a Web application's web.xml file for a resource and hitting the protected resource with HTTP, the server threw an error saying the user was not authorized to view the resource. The server did not give the user a chance to authenticate and then switch over to HTTPS according to servlet specifications 2.3 page 87. The container now redirects the client to the HTTPS port if the original request was over HTTP and resources are marked integral or confidential.
CR092428	WebLogic Server did not accept "," as a character in Netscape cookies. A comma is now allowed for Netscape cookies because commas can be used as delimiter only for RFC2109 cookie.
CR092545	If a request is issued through the proxy, and then the same request is issued again without letting the first finish, an exception occurred. The proxy now drains the inputstream from IIS even when it has failed to write to the client (connection was closed by the browser).

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.setCharacterEncodin g(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.
CR093014	Java comments spanning multiple scriptlets caused `jspc` to fail.
	Adding code that causes the `ScriptletScopeLexer` to skip an entire Java comment solved the problem.
CR093112	If a JSP with a line such as `<%@ include file="./file.inc" %>` and `file.inc` contains a line such as `<%@ taglib uri="/c.tld" prefix="c" %>` and `c.tld` has a validator, a JSP Compiler error with `<%@ include file="./file.inc"%>` and using tag lib was returned.
	This problem was fixed by adding a reference to the currentURI of the JSP page being translated into XML format. This allows the implementation to calculate the location of the included JSP relative to the application root.
CR093167	When using the HTTPClusterServlet to access a secure Web application on a backend cluster, Internet Explorer will 'hang' for 30 seconds after the username and password has been entered.
	WebLogic Server now closes the connection between the client and the proxy server if a Connection: Close Header request has been proxied to the browser.
CR093209	A NullPointerException was thrown from WebLogic Server when a Web application was accessed (after successful deployment) which had a property set to java protocol in its application.
	The code now checks for null value of `source.getURL()`.

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(ServletS
tubImpl.java:262)
at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:198)
at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(Web
AppServletContext.java:2637)
at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequ
estImpl.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()`.

CR093625	The 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.

CR093634	A new parameter, Weblogic Plug-In Enabled, was added to WebLogic Server. A call to `getRemoteAddr` returned the IP address of `HttpClusterServlet`, and not the IP address of the client, because `HttpClusterServlet` did not set the proprietary header, "WL-Proxy-Client-IP" and therefore used the socket's IP address.
	This has been corrected in the code.

CR093755	Innocuous warning message removed from code.
	context.log.error("Warning: One of the getParameter family of methods called after reading from the ServletInputStream(), can't mix these two!"); was turned off
CR094190	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 __jspService method.
	The problem was solved by a code change. Now, when an empty BodyTag of the form <my:tag ... /> 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 StandardTagLib.java that tricks the BodyTag into thinking it is being invoked as usual from the JSP source.
CR094416	In earlier versions of WebLogic Server 7.0, WebLogic Server precompiled all JSPs, even those that had not been updated.
	A code change now prevents the temp directory from being overwritten when an instance of WebLogic Server is restarted.
CR094488	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 AuthCookieEnabled="true" to the WebServer element in config.xml:
	<WebServer Name="myserver" AuthCookieEnabled="true"/>
	This causes an 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.
	Note: If authenticating via plain HTTP, the secure cookie is not set or required for any HTTPS resources. When accessing a non-protected HTTPS resource, the cookie is not verified (since it will not have been sent from the browser). This behavior allows the browser to access non-protected HTTPs resources without the user logging in.
CR094663	When running a load test with a 3 server cluster and using IIS proxy plugin, then shutting down one of the servers in the cluster, the IIS proxy threw a 500 error.
	This was due to both a parsing error and a server affinity problem.
	These problems have been fixed in the code.
CR095713	Unix platforms (Solaris, Linux) showed loss of replicated sessions on secondary servers.
	The session-config element in web.xml (j2ee descriptor) is not deprecated. The value is in minutes. The weblogic.xml (BEA-specific descriptor) element takes a value in seconds and takes precedence over the value defined in web.xml.

CR096399	JSP compilation errors were handled improperly. The filter communicated with `OutputStream` instead of with `HttpServletResponseWrapper getWriter`, with this result:

```
java.lang.IllegalStateException: strict servlet API: cannot call
getWriter() after getOutputStream() at
weblogic.servlet.internal.ServletResponseImpl.getWriter(ServletR
esponseImpl.java:171) at
weblogic.servlet.internal.ServletRequestImpl.reportJSPFailure(Se
rvletRequestImpl.java:227) at
weblogic.servlet.internal.ServletRequestImpl.reportJSPCompilatio
nFailure(ServletRequestImpl.java:239) at
weblogic.servlet.jsp.JspStub.reportCompilationFailure(JspStub.ja
va:523) at
weblogic.servlet.jsp.JspStub.compilePage(JspStub.java:430) at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:210) at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:164) at
weblogic.servlet.internal.ServletStubImpl.getServlet(ServletStub
Impl.java:517) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:351) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:445) at
weblogic.servlet.internal.TailFilter.doFilter(TailFilter.java:20
) at
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainIm
pl.java:27) at WebLogic ServerBugFilter.doFilter(WebLogic
ServerBugFilter.java:27) at
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainIm
pl.java:27) at
weblogic.servlet.internal.WebAppServletContext$ServletInvocation
Action.run(WebAppServletContext.java:5418) at
weblogic.security.service.SecurityServiceManager.runAs(SecurityS
erviceManager.java:744) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(Web
AppServletContext.java:3086) at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequ
estImpl.java:2544) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)...

WebLogic ServerBugFilter.doFilter after
```

The problem was fixed by the implementation of a wrapper-aware `RequestCallback`.

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(Itera`
	`torEnumerator.java:25) at`
	`com.colibria.core.xmlswitch.XMLSwitch.getQueryStringXML(XMLSwitc`
	`h.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 with 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.
CR098366	Right after server startup some server requests were rejected because the MBean did not update server state before the listening thread started receiving requests.
	This problem has been resolved by using Boolean flags inside the Web container to update the server state for the listening thread.
CR098518	The following EJB/Web application setup was causing a class cast exception:
	`Servlet Filter1 --> WebApp1 -- forward to WebApp2 ctx --> Servlet` `Filter2(WebApp2) --> Ejb Lookup (fails)`
	This problem has been fixed by locating the EJB lookup in the Servlet instead of a Servlet Filter when doing a forward between two Web applications.

CR100172	When multiple chunked requests are posted to a servlet a number of the requests fail with a `NumberFormationException`. Mostly of the time alternate requests are fulfilled and the next request fails. It seemed to be occurring when the requests were being made on the same connection. The exception stack trace was:
	
	`<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(HttpChunk`
	`InputStream.java:118) at`
	`weblogic.utils.http.HttpChunkInputStream.initChunk(HttpChunkInpu`
	`tStream.java:72)`
	
	Analysis revealed two problems in `PostInputStream`:
	• It did not reset control variable readAllChunks when recycling PostInputStream
	• `isChunkComplete` did not detect the end of chunk correctly
	
	The problem was resolved by a code fix to read until end of stream in PostInputStream and correctly detect end of chunk in HttpChunkInputStream.

CR100298	A Web service using a filter servlet that, in turn, used `HTTPRequest` wrapper classes on top of an `HTTPServletRequest` resulted in a `ClassCastException` error, as WebLogic Server was expecting that request to be of the type `weblogic.servlet.internal.ServletRequestImpl`. A code fix resolved the problem.

CR100572	When a request with an incorrect URI was received from a from plug-in, WebLogic Server threw the following stack trace:
	
	`<Mar 8, 2003 3:27:20 PM PST> <Error> <Socket> <BEA-000421>`
	`<Uncaught Throwable i n processSockets`
	`java.lang.NullPointerException.`
	`java.lang.NullPointerException...`
	
	This problem was solved with a code fix.

CR100837	WebLogic Server 6.1 SP04 misdirected output for JSP `includes` in JSP pages. The problem was corrected with a code fix.
CR102036	The first access to deleted JSP pages was causing a `JspFileNotFoundException` instead of 404 File Not Fount. Further access to this page correctly returned 404. The problem is fixed so that deleted JSP pages always return 404.

CR102446   In previous versions of WebLogic Server 7.0, if you invoked a JSP from a Web application running on an Admin Server so that the JSP registered a `ModelMBean` on the Admin Server `MBeanServer`, and then stopped and restarted a Managed Server, the server would fail with the following message:

```
<Critical> <WebLogicServer> <000364> <Server failed during
initialization.
Exception:weblogic.management.configuration.ConfigurationExcepti
on:  - with nested exception:
```

```
[java.lang.ClassCastException:
javax.management.modelmbean.ModelMBeanInfoSupport]
```

```
java.lang.ClassCastException:
javax.management.modelmbean.ModelMBeanInfoSupport at
weblogic.management.commo.Commo.createMBean(Commo.java:530) at
weblogic.management.internal.RemoteMBeanServerImpl.createCommoMB
eanLocally(RemoteMBeanServerImpl.java:941) at
weblogic.management.ManagedServerAdmin.syncToAdminServerCommo(Ma
nagedServerAdmin.java:406) at
weblogic.management.ManagedServerAdmin.initializeCommo(ManagedSe
rverAdmin.java:444)
        at weblogic.management.Admin.start(Admin.java:307)
        at weblogic.t3.srvr.T3Srvr.initialize1(T3Srvr.java:679)
        at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:589)
        at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:277)
        at weblogic.Server.main(Server.java:32)
```

Code changes in `weblogic.management.commo.Commo.java` resolved the problem.

CR102628   The following JSP code:

```
<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/plugin-in
stall.html";
iepluginurl="http://java.sun.com/products/plugin/1.1.3/
jinstall-113-win32.cab#Version=1,1,3,0" >
```

generated this HTML:

```
<embed type="application/x-java-applet;version=1.3.1_06"
pluginspage="http://java.sun.com/products/plugin/1.1.3/plugin-in
stall.html"; height="800" width="500"
code="examples.applets.PhoneBook1.class" codebase="/bea_WebLogic
Server_internal/classes/DefaultWebApp@DefaultWebApp/">
```

The generated HTML code failed on Netscape: Netscape attempted to download the plug-in from WebLogic Server instead from the SUN site.

When the `pluginspage` line was moved before the `codebase` line, the code worked correctly on Netscape.

A code fix to order the applet attributes properly resolved the problem.

CR102675	If a taglib is imported once via the taglib directive in a page that includes a second page, and that second page also imports that same taglib, then invalid XML results in the following browser output:

CR102675

If a taglib is imported once via the taglib directive in a page that includes a second page, and that second page also imports that same taglib, then invalid XML results in the following browser output:

```
Error 500--Internal Server Error From RFC 2068 Hypertext Transfer
Protocol -- HTTP/1.1:
```

```
10.5.1 500 Internal Server Error The server encountered an
unexpected condition which prevented it from fulfilling the
request.
```

And the following log file output:

```
<[ServletContext(id=287644,name=jspVal.war,context-path=/jspVal)
] Servlet
```

```
failed withIOException java.io.IOException:
javax.servlet.jsp.JspException: The taglib validator
```

```
rejected the page: "Exception TestValidator XML parsing:
org.xml.sax.SAXParseException:
```

```
Attribute "xmlns:valtest" was already specified for element
"jsp:root". -- stack trace written to stderr., "...
```

Because of a code change to `weblogic.servlet.jsp.jsp2xml.Jsp2XmlOutputter`, duplicate tag prefixes are ignored in the generation of the XML views of the JSPs.

CR102769

Forwarding a request from a servlet or JSP to another JSP page did not log `queryString` in the `access.log`. Additionally, `access.log` logged the forwarded request, but not the original request.

A code change to `RequestDispatcherImpl` fixed the problem.

CR103289

The `HttpClusterServlet` was not correctly parsing the session id from post data.

If you sent a request through `HttpClusterServlet` 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.

The servlet is now able to extract the session id from the post data.

CR103473

Performance improvements were made to the command-line compilation of `jspc`; One compiler process is now forked for each JSP, rather than multiple compiler processes. In addition, a TLD used for multiple JSPs is now parsed just once for all JSPs, rather than once each.

CR105395

If a JSP called Security.getCurrentUser() with no user logged in, the following exception resulted:

```
java.lang.NullPointerException at
weblogic.security.acl.Security.getCurrentUser(Security.java:301)
```

The code was changed to tolerate wlrealm null settings without throwing an exception.

JTA

Change Request Number	Description
CR088171	In previous releases of WebLogic Server 7.0, an unexpected exception was thrown while committing or rolling back a server-initiated transaction when the coordinating server was not the initiating server. The subordinate initiating server that was waiting for commit/rollback processing to complete was interrupted by an unexpected exception, often a `PeerGoneException` resulting from a restart of the coordinating server, and the transaction status was changed to "unknown." When the coordinating server restarted, it was not able to resolve the transaction correctly. A code change allows the coordinating server to resolve the transaction correctly after restart.
CR091628	The `javax.transaction.xa.XAResource` interface defines a method, `setTransactionTimeout()`, which 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. The following properties were added to the JDBCConnectionPool object in the config.xml file: • `XASetTransactionTimeout`—Determines whether the Transaction Manager will call `XAResource.setTransactionTimeout` before the call to `XAResource.start`. Possible values are `true` and `false`. When set to `true`, the Transaction Manager passes the global transaction timeout in seconds to Oracle as the Session Timeout. When set to `false`, the Transaction Manager does not call `XAResource.setTransactionTimeout`. The default value for this parameter is `false`. • `XATransactionTimeout`—When `XASetTransactionTimeout` is set to true, the `XATransactionTimeout` value is sent to Oracle as the Session Timeout (same as setting the `SesTm` in an OpenString). When this parameter is set to `0` the Session Timeout is set to the global transaction timeout in seconds. The default value for this parameter is `0`. These parameters are valid for connection pools that use an XA JDBC driver to create database connections. They are ignored for connection pool that use a non-XA JDBC driver. These parameters are not available in the Administration Console.
CR091882	In previous releases of WebLogic Server 7.0, a problem can occur when a transaction spans multiple WebLogic Server instances. With some configurations it is possible that transactions can accumulate redundant state resulting in increased completion times and possibly leading to transaction time outs. This problem was fixed by a code change.

Change Request Number	Description
CR091925	In previous releases of WebLogic Server 7.0, a problem could occur with the recovery migration service with clustered Managed Servers when an exception was not caught. This could cause Managed Server boot failures when other members of the cluster were not yet running. This problem was fixed by a code change.
CR092301, CR107866	In previous releases of WebLogic Server 7.0, a `NullPointerException` exception can occur during transaction coordination for some configurations. This problem was fixed by a code change.
CR093406	In WebLogic Server 7.0, transaction recovery processing on server restart was delayed for newly discovered resources. In WebLogic Server 7.0 SP3, WebLogic Server begins transaction recovery processing on a resource as soon as it is discovered by a server.
CR097013	A subcoordinator server was not recovering in-doubt transactions automatically after server down, requiring manual rollback or commitment by the DBA. Research revealed that the subcoordinator code was losing track of remote participants after server restart. A checkpoint was added to the code to properly track remote participants.
CR098273	In WebLogic Server 7.0, stale entries in the transaction log sometimes caused unnecessary recovery overhead when a server is restarted. This problem was solved with a code fix.
CR098318	In WebLogic Server WebLogic Server 7.0 SP1, a stale transaction state can become associated with dedicated communication threads which can then erroneously infect remote servers with a transaction, resulting in `java.lang.IllegalStateException` exceptions being generated. This problem was fixed by a code change.
CR099554	In WebLogic Server 7.0 SP1, under certain circumstances when a server with JMS messages in a pending state is shut down or crashes, pending messages are not recovered when the server is restarted. The problem was resolved with a code fix.

Change Request Number	Description

CR099830	JTA Migration sometimes failed with a `NullPointerException` or `ConnectException` after a transaction was propagated among two servers in cluster.
	
	`NullPointerException <Feb 28, 2003 6:38:44 PM JST> <Warning> <JTA>`
	`<110213> <The activation of Transaction Recovery Service for`
	`server [serverB] fails. java.lang.NullPointerException at...`
	
	`ConnectException <2003/02/27 13:11:20:JST> <Warning> <JTA>`
	`<110213> <The activation of Transaction Recovery Service for`
	`server [server1] fails.> java.rmi.ConnectException: Destination`
	`unreachable; nested exception is: java.net.ConnectException:`
	`Connection refused: connect; No available router to destination`
	`at...`
	
	Research revealed that the code created a race condition between the notification of when the server was dead and when the information for that server was cleaned from the cache. If the notification that the server was dead came after the cache was cleared, the remaining Managed Server would get a NULLPointer Exception when it tried to look for the location of the tlog of the dead Managed Server.
	
	A code fix resolved the problem.

CR100898	If a transaction was rolled back, and a server participant had no resource participants, it was possible that the transaction would never be completed. The transaction continued to exist on the coordinating server and continually retried notifying the server with no resources to roll back. Eventually, the transaction was 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 asynchronous reply processing to accommodate servers with no resources and to immediately set state to rolled back when all servers and resources are accounted for.

Change Request Number	Description
CR101135	As part of transaction timeout processing, for each transaction that had been prepared and for which the current server is a subordinate participant in the transaction, the Transaction Manager will send a one-way RMI message to the coordinator to determine the status of the transaction. If the coordinator was restarted, it is possible for this status request to be dispatched before the JTA subsystem finishes reading the commit records from the transaction log.
	In this case, the status request was dispatched and checked to see if the transaction IDs being queried exist in the Transaction Manager's map of known transactions. Because the tlog had not yet been processed, the TM responded with a rollback decision to the subordinate, even if the transaction had passed the commit point prior to the coordinator restart.
	This scenario resulted in a mixed transaction outcome where the resources on the subordinate making the status request are rolled back while other resources in the transaction are committed.
	The code was changed so that it does not process checkStatus() requests before the JTA subsystem has initialized. This prevents a mixed transaction outcome.
CR106174	The Transaction Manager was unnecessarily writing a commit record when all but one participating resource in a transaction respond to the prepare command with the XA_RDONLY flag.
	A code change resolved this problem. Now, before writing a log record, the server checks to see how many participating resources require second phase. If the number is one, the transaction does not generate a commit record.
CR106174	If all but one participating resources in a transaction respond to prepare with the XA_RDONLY flag, the transaction manager should issue a one-phase commit to the pending participant. However, the Transaction Manager was unnecessarily writing a commit record for this scenario.
	The code was changed so that, before writing a log record, the Transaction Manager checks to see how many participants require second phase; if one does, the write is skipped. This resolved the problem.
CR106177	While processing remote beforeCompletion callbacks, the Transaction Manager resumed the transaction context before invoking the registered objects. Statically registered resources were not enlisted when the transaction was resumed on the thread, which prevented beforeCompletion logic to perform additional updates as part of the transaction.
	The Transaction Manager now allows enlistment of statically registered resources.

Change Request Number	Description
CR106177	While processing remote beforeCompletion callbacks, the TM resumes the transaction context prior to invoking the registered objects. However, statically registered resources are not enlisted when the transaction is resumed on the thread.
	The code was changed so that Enlistment is performed on such resources to allow beforeCompletion logic to perform additional updates as part of the transaction. This resolved the problem.

Operations, Administration, and Management

Change Request Number	Description
CR098464	LogManager's use of synchronization was causing a deadlock in the threads.
	Eliminating some synchronization in LogManager fixed the problem. There is no more deadlock.

Plug-ins

Change Request Number	Description
CR096850	The iPlanet server no longer crashes after an application (including a load testing application) makes multiple attempts to connect.
CR086224	The 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.

CR091791	When the Apache plug-in received a new dynamic server list from the backend server it reset the keepAliveSecs to 0. This resulted in the timer removing all the connections from the pool. So effectively connection pooling was broken.
	A code fix eliminated this problem.
CR092484	Dynamic server list was being parsed incorrectly with a cluster, by the Apache plug-in.
	This problem was resolved and the server list is now being parsed correctly for versions going back to 5.1.
CR092756	ISAPI plugin was not failing over on a HTTP 503 response from the backend WebLogic Server.
	A code fix caused the plug-in to fail over when it receives a HTTP 503 response from WebLogic Server.
CR093530	`PathTrim` value was case sensitive, and needed to be made case insensitive, for the ISAPI plug-in.
	A code fix provides case insensitivity to the PathTrim value.
CR093595	When using a long URI and a weblogiccluster instead of weblogichost, the Apache plug-in was shutting down unexpectedly.
	A code fix increased the buffer size to the limit of the URI length, which resolves this problem.
CR094768	Parsing post for cookies was not implemented correctly in the ISAPI plug-in.
	The ISAPI plug-in now parses the post only if the content-type is `application/x-www-form-urlencoded`.
CR096625	The `X_WEBLOGIC_FORCE_JVMID` header was sent to a server instance in a cluster, by the Apache plug-in.
	`X_WEBLOGIC_FORCE_JVMID` should only be sent when the server list is non-clustered and the current server does not have a `jvmid` yet.
	This problem was solved by a code fix.
CR097132	Apache plug-ins needed to be created for Linux-IA64.
	The binary files and libraries have been created, and the new plug-in is available.

CR097202 CR103161	A customer needed 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 the following code to obtain the browser's cipher strength:
	
	`(Integer)httpRequest.getAttribute("javax.servlet.`
	`request.key-size")`
	
	However, the value returned was always 128 (although the browser was set to use a cipher size less than 128-bit).
	
	This problem occurred because `HTTPS_KEYSIZE` determines the strength of the connection that is made. Because 128-bit certs were used on the plug-in, 128 was returned, regardless of the strength used by the browser. When the request was directed to WebLogic Server, 40 or 128 was returned, depending on the browser strength. The plug-in does not modify the `HTTPS_KEYSIZE` or `HTTPS_SECRETKEYSIZE` headers.
	
	To meet this need, two new headers were implemented: `WL-Proxy-Client-Keysize` and `WL-Proxy-Client-Secretkeysize`. Both headers values can be obtained using `request.getHeader()`.

CR100361	The following new exception types were added for the NSAPI plug-in:
	`READ_ERROR_FROM_CLIENT, READ_ERROR_FROM_SERVER,`
	`READ_ERROR_FROM_FILE, WRITE_ERROR_TO_FILE`

CR101428	With the ISAPI plug-in, when requests contained multiple cookies, cookies not named JSESSIONID but that immediately followed the JSESSIONID cookie were stripped off and not delivered to WebLogic Server. The problem was solved with a code change.
CR101596	In the `ssl_certchain_verify_callback` method, `SSLNoErr` was returned as the error condition for some failure cases of certification validation. This included cases where the CA certificate had basic constraints that were not marked as critical and the strict setting was enabled (instead of the default "strong" setting), and if the certificate was not marked as being a CA. This problem was resolved by returning X509CertChainInvalidErr for these failure cases (instead of SSLNoErr).
CR102616	The NSAPI 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 WebLogicHost = 'DNS name'.

RMI/RMI-IIOP

Change Request Number	Description
CR108317	Some accessors did not map properly in the ejbc-generated IDL (Interface-Definition-Language). The problem has been resolved.
CR090979	When setting principal and credentials in a client that then contacts a server that contacts another server, the server-to-server credential propagation of principals was not correct. A code fix provides correct credential propagation.

CR090987	WebLogic failed to get an ROID from a server and return it to a client, and threw the following exception:
	
	`java.rmi.UnmarshalException: failed to unmarshal class`
	`weblogic.cluster.replication.ROID; nested exception is:`
	`java.rmi.UnmarshalException: null; nested exception is:`
	`org.omg.CORBA.MARSHAL  vmcid: 0x0  minor code: 0  completed: No`
	
	A code fix properly marshals the class.

CR093107	The clustered reference used by CosNaming contains a single concrete reference, causing IIOP failover handling to malfunction at the JNDI level. The reference was not being bound into JNDI. A code fix resolved this problem.
CR093561	A code fix improved replacement for clusterable objects.
CR094795	A fat client connects to a stateless EJB and then invokes a method that returns a class that extends Vector. The class extending Vector uses its own serialization to improve optimization of the serialization. The fat client displays the contents of this Vector class. If RMI-IIOP is used as the protocol, then the Vector is empty. Analysis revealed that the call to `writeReplace` was too late. The repository ID had already been written so that WebLogic Server read the actual class rather than the class that should be resolved. A code fix invokes `writeReplace` properly for externalizable classes.

Change Request Number	Description

CR096476	The MBean property `Server.IIOP.IdleConnectionTimeout` configures how long the server lets a connection be idle before closing it. WebLogic Server did not have a similar client-side property.
	
	A code fix implemented an idle timeout and a pending timeout. The pending timeout is controlled by the DGC period length and number.
	
	To set the idle timeout to 100s (default is 240 s):
	`-Dweblogic.iiop.IdleConnectionTimeout=100`
	
	To set the pending timeout to 5 minutes (default is 60 s):
	`-Dweblogic.DGCIdlePeriodsUntilTimeout=5`
	
	To set the period length:
	`-Dweblogic.PeriodLength=120000` (for 120 s)
	
	A connection is be timed out if the connection idle period has not elapsed. It is assumed that the pending timeout is always greater than the idle timeout. The pending timeout is used for outbound connections that are still awaiting responses.

CR098713	WebLogic Server does not support outbound chunking of IIOP valuetypes, which meant it did not support change to externalizable classes. A code fix implemented support for outbound chunking.
CR099693	In 7.0 (G.A., SP01, and SP02), a client accessing WebLogic Server through a firewall using the t3 protocol received an error because the server expected to route a message over a connection that was not initialized. The server's `ExternalDNSName` was specified. Upon initial connection, the client addressed the request to the server instance's `ExternalDNSName`. The JVMID in the response from the server instance requires the server's internal IP address and internal DNS name. Subsequent requests from the client were not recognized. A code fix resolved the problem.
CR100480	A code fix correctly maintains IIOP client context IDs (including the thin-client) for multiple threads doing competing authentications.

Change Request Number	Description
CR100831	When the client was on a server in one cluster and was using a replica-aware stub for a server in another cluster, the local server could not be in the replica list, and therefore, the preferred host was ignored.
	This problem was resolved by modifying the code to check for local affinity (which will never be obtained when calling a remote cluster). If the local affinity cannot be obtained, then the preferred host is compared against the replica list. If and only if there is no match in the replica list will a host be chosen from the list using the load-balancing algorithm.
CR101186	A bug prevented exception interoperability working over IIOP. Specifically an exception thrown by a server running on JDK 1.4.x (i.e 8.1) could not be understood by a 7.0.x server running on JDK 1.3.1. This manifested itself as a MARSHAL exception. There were two bugs, one in the handling of sequences and the other in the way classes with described by FVDs (FullValueDescriptions). Exception interoperability now works over IIOP.
CR103388	WebLogic Server RMI now supports arrays of primitives.
CR107137	Over CORBA and IIOP, client-server access generated marshalling and indirection errors.
	Code changes fixed the generation of hashCodes during indirection, and also equals for IndirectionWrapper.

Security

Change Request Number	Description
CR080488, CR091569	A sample custom Authentication provider that supports the user/group management Security Service Programming Interfaces (SSPIs) was included in the sample security providers code example on dev2dev. In addition, a custom Auditing provider was added to the code example. For more information, see http://dev2dev/codelibrary/code/security_prov81.jsp.
CR084874	Within two trusted domains, a Java client calls stateless session bean1 on server1 in domain1. This bean calls stateless session bean2 on server2 in domain2. The user calling the first server is user `system` in group `Administrators` with Role `Admin` (default). If the call from bean 1 to bean 2 used the t3 protocol, the principal was propagated properly. If the call from bean 1 to bean 2 was an IIOP call and there was a security role assignment in the `weblogic-ejb-jar.xml` that mapped the role `Admin` to `user` system, the principal was propagated properly. However, if the call from bean 1 to bean 2 was an IIOP call and the security role assignment mapped role `Admin` to group `Administrators,` an `org.omg.CORBA.NO_PERMISSION` error occurred. The principal propagation now works properly in all situations.
CR085094	After creating a domain with one Administration Server and one Managed Server, enabling the `-Djava.security.manager` for the two servers returned a `java.io.Filepermissions` error reported in the AdminServer log for `<Error> <EmbeddedLDAP> <000000> <Error parsing Entry #19: access denied (java.io.FilePermission .\myserver\ldap\ldapfiles\EmbeddedLDAP.data read)>`. This problem was resolved by updates to the `weblogic.policy` file, to which was added information about granting permissions to user applications, and examples of grant statements. The `weblogic.policy` works for Weblogic Server Examples, Pet Store, and Workshop. It needs to be modified to be used for user configurations; instructions are in the comments at the top of the file.

Change Request Number	Description
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 won't pass the certificate validation, an information message is logged noting that clients could reject it. For more information, see SSL Certificate Validation.
CR090147	When using the Active Directory LDAP Authentication provider, the user search filter did not support the use of backslashes in a user name. The *LDAPv3 Attribute Syntax Definition* specification supports the use of backslashes in the user search filter. The code for the user search filter has been updated to support the use of backslashes.
CR090218	The methods `getCurrentUser` and `getCurrentSubject` are supposed to return the user ID of the user associated with the current thread. In previous Service Packs of WebLogic Server 7.0, if you forwarded from one servlet to another using `RequestDispatcher.forward`, these methods would return null (or an empty string). A code fix resolved this issue so that requests forwarded from one JSP or servlet to another will carry the authenticated user in the session.
CR091972	In previous releases of WebLogic Server, the SSL license check required both the 128 bit and the IP address using the license. The original intent of the license check was: • To ensure customers could not use the Certicom SSL runtime except when running WebLogic Server. This was a requirement of BEA System's agreement with Certicom. • To determine what strength SSL could be used (domestic or export). This was a government regulation. BEA recommended that Java clients not running on a server copy the license from the server to the Java client machine. However, the implementation of the license check in the SSL code asked for an IP validation. Therefore, the server SSL license could not be used on the client machine. The IP validation in the license check has been removed. The license can now be copied from the server to the Java client machine.
CR092236	When using the Active Directory LDAP Authentication provider, the user search filter did not support the use of nature language characters in a user name. The *LDAPv3 Attribute Syntax Definition* specification supports the use of natural language characters in the user search filter. The code for the user search filter is updated to support the use of nature language characters.

Change Request Number	Description
CR093533	The Realm Adapter Authentication provider in the Compatibility realm used the authenticated user name in the User object instead of creating a User object with a unique ID. This problem broke backward compatibility with custom security realms that perform simple authentication (username/password).
	The Realm Adapter Authentication provider code has been modified to create a User object with a unique ID.
CR093813	The password used to access the encrypted private key in the Node Manager key file, `weblogic.nodemanager.keyPassword`, was in plain text and accessible.
	The problem was resolved by the creation of the Node Manager properties file, `nodemanager.properties`.
CR094038	The hierarchy caching optimization has been removed from the resource engine within the WebLogic security framework. The resource cache has been updated to use Resource as a key. The cache now runs `resource.equals()` on every cache hit to make sure there is not a hash code collision. The resource string representation and parent are cached in the resource, so after the hierarchy is in cache only one cache lookup is required to get the resource name path. This change improves the performance of WebLogic Server.
CR094215	A call to `weblogic.security.ntrealm.NTRealm` *username password* always raised an exception. This problem has been resolved.
CR094548	In earlier versions of WebLogic Server 7.0, groups in LDAP had to be explicitly added to one another for WebLogic Server to recognize that the two groups were nested. Implicit nesting, by adding a group under another group in an LDAP tree, was not recognized.
	This problem has been resolved by a modification to the LDAP provider that makes it perform group membership checking by hierarchy. Set the property as follows:
	`-Dweblogic.security.hierarchyGroupMemberships="true"`
CR095689	When two-way SSL and the `weblogic.servlet.security.ServletAuthentication.strong()` method were used, an exception occurred. The import in the authentication portion of the servlet code used `javax.security.cert.X509Certificate` instead of `java.security.cert.X509Certificate` as defined by the servlet specification. The servlet code now uses `java.security.cert.X509Certificate`.

Change Request Number	Description

CR095789	The LDAP Realm v2 had poor performance when handling groups with a large number of users. This problem could cause the WebLogic Portal server to hang and time out. The problem was related to the calls to:
	
	`netscape.ldap.LDAPMessageQueue.waitForMessage`
	`(LDAPMessageQueue.java:179)`
	
	Specifically, the calls hung if there were many groups in the `groupDN` were searched by the LDAPRealm v2 and the `Group.isMember` method was called in succession for all of these groups. Generally, the server would hang after several hundred calls in a row.
	
	The code for the LDAP Realm v2 was updated to resolve this problem and improve performance when handling groups with large numbers.

CR095836	The default value for backup minutes is now correctly calculated and reported under domain > Domain Wide Security Settings -> Embedded LDAP in the Administration Console.
CR096488	When using a 6.1 security configuration that uses the RDBMS security realm, the server (and therefore Compatibility security) would not start. 1. The RDBMS Realm `config.xml` entry needs to explicitly state the RealmClassName, Database Driver and Database URL attributes. When using the RDBMS security realm in 6.x, these values assumed the RDBMS code example and the examples database were used. 2. The HimselfAAA code must be recompiled to work with WebLogic Server version 7.0. In WebLogic Server 6.x, a private Admin API was exposed in the RDBMS code example. This is a known issue. For more information, see Upgrading Security. 3. The Weblogic Server code which loads the security realms did not properly defend against cases where the RealmClassName attribute would be empty. This code has been updated and an exception which clearly states that a Realm Class Name must be supplied is thrown.
CR096589	`weblogic.servlet.security.ServletAuthentication.strong()` was failing with a class cast exception. Research revealed that the wrong certification class was being imported. A code fix resolved the problem.
CR096934	Construction of an `AuthenticationProvider` was not recognizing errors, because `WebLogicBeanMaker` was not seeing exceptions thrown by the `AuthenticationMBI` class constructor. The problem was resolved by a code change ensuring the propagation of errors.

Change Request Number	Description
CR098083	In past releases of WebLogic Server, the WebLogic Credential Mapping gave no notification when it received passwords that it could not validate. This problem has been resolved. The WebLogic Credential Mapping provider now throws an exception when it receives encrypted passwords that cannot be validated.
CR098084	In past releases of WebLogic Server, the passwords stored in the embedded LDAP server by the WebLogic Credential Mapping provider were in clear text. The WebLogic Credential Mapping provider now encrypts passwords when storing them in the embedded LDAP server.
CR098242	Node Manager allowed unauthorized starting of Managed Servers, because it supplied a cleared user name and password regardless of the authorization level of the user. Node Manager now performs a security check on the user.
CR100174	The Bouncy Castle JCE provider was not tested prior to 7.0 SP3, and requires special case code to get it to work with SSL. In this release, code was added to SSL to detect when the BouncyCastle provider JCE is plugged in, and fall back to use internal cryptography instead. Customers that want to use the BouncyCastle JCE provider in their server side applications can do that now without provoking an exception. SSL will not use that provider if it is plugged in, but the exception will not occur.

CR100592	When a user account in one of the supported LDAP servers became disabled (for example, if the account was disabled for a long period time or locked because of a failure), any attempt to use that account (username/password) from an LDAP security realm resulted in the need to restart the server. The following exception was throw:
	
	 `<[WebAppServletContext(2052089,WebCM-Outages,/Webbed-Outages)]`
	 `Servlet failed with Exception netscape.ldap.LDAPException`
	
	 This problem has been resolved.

Change Request Number	Description
CR100762	In previous releases of WebLogic Server, the code that unloaded the SunJCE provider at runtime was still being used. This problem did not allow the use of the Sun JCE provider with WebLogic Server when the SSL protocol was enabled on the server instance.
	This problem has been resolved.
	Customers who are using JDK 1.4 clients with Sun JCE may experience a performance degradation due to a performance issue in the JDK 1.4 JCE implementation. If the client does not require the Sun JCE provider, you should comment out the Syncs provider from the `JDK/re/lib/security/java.security` file as follows:
	``` security.provider.1=sun.security.provider.Sun security.provider.2=com.sun.net.ssl.internal.ssl.Provider security.provider.3=com.sun.rsajca.Provider #security.provider.4=com.sun.crypto.provider.SunJCE #security.provider.5=sun.security.jgss.SunProvider ```
CR100797	The default connection filter in WebLogic Server was not handling the HTTPS protocol properly. This problem has been resolved.
CR101652	In previous releases of WebLogic Server, it was not possible to make a secure connection to a third-party XML server because the XML server did not support the SSL protocol.
	The following workaround is provided:
	1. Create a WebLogic client (a Java Server page (JSP) deployed in WebLogic Server).
	2. Create an Java authenticator class according to the Java specification. This authenticator should obtain a surname and password for the used to connect to the XML server.
	3. Specify Net Permission for the Java authenticator class in the `weblogic.policy` file. The Net Permission is granted to all users of that Java VM. Note this presents a possible security weakness and should therefore be used carefully.
	4. Use `java.net.HttpConnection` to establish a connection to the XML server.
	5. Specify `-DUseSunHttpHandler` as a command-line argument when starting WebLogic Server.
CR102351	Eracom and Entrust third-party security providers have been certified for this release.

Change Request Number	Description
CR102351	7.0 SP2 enabled JCE providers to be used for SSL cryptography operations. Unfortunately, all JCE providers do not behave the same way; each requires special handling and foreknowledge. The ERACOM/Entrust JCE providers require special case code to work with SSL.
	Code added to SSL causes it to detect when the ERACOM/Entrust providers are plugged in, and to use internal cryptography. ERACOM/Entrust JCE providers work in server-side applications now.
CR105809	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-33.jsp.
CR106027	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp.
CR106357	When multiple authorization providers were installed, you could not define policy on Web applications.
	This problem has been fixed.

System Administration

Change Request Number	Description
CR063279	When multiple elements in WebLogic Server's `config.xml` had the same objectnames, later elements overwrite configurations for preceding elements. This used to happen silently. Now, WebLogic Server throws an error message (ID=150012) to standard output.
CR082422	In Unix, newly deployed files did not display in file or directory listings unless you changed to the root directory and then changed back to the newly deployed file's parent directory.
	This problem was resolved by code changes to `file_chooser.jsp`.
CR091205	The WebLogic Server SNMP MIB had a duplicate OID (1.3.6.1.4.1.140.625.105.1.1) for `applicationRuntimeIndex` and `cacheMonitorRuntimeIndex`. `cacheMonitorRuntime` now uses a unique OID: 1.3.6.1.4.1.140.625.107.
CR091224	Fixed a problem where calling `EJBComponentRuntimeMBean.getEJBRuntimes()` caused an `ASSERTION FAILED` error followed by an `ArrayStoreException`.

Change Request Number	Description
CR092877	The `ServerLifecycleRuntimeMBean` and the `ServerConfigMBeans` for a deleted Managed Server continued to exist until the Administration Server restarted. Deleting their Managed Server now deletes these MBeans.
CR092946	Fixed a bug that caused the following exception when starting the Administration Server and using SNMP: `java.lang.ClassNotFoundException:` `weblogic.management.snmp.agent.VirtualHost`
CR093481	Fixed a problem that displayed an unnecessary error message if both `weblogic.Domain` and `weblogic.apache.xerces.maxentityrefs` are set when starting WebLogic Server.
CR093816	Fixed a problem causing a NPE when using dynamic MBeans.

CR093824	Fixed a problem when restarting an Administration Server that caused the following exception when using the domain-wide Administration Port. This exception does not occur when the domain-wide Administration Port is *not* used, nor is the exception thrown when the domain is initially booted:
	
	 `<Emergency> <Configuration Management> <150009>`
	
	 `<Errors detected attempting to connect to admin server ...`

CR094966	Fixed a problem that causes application deployment to fail after restarting an Administration Server while Managed Servers are running.
CR096544	On HP-UX, migration to 7.0 from earlier versions of WebLogic Server sometimes failed with an `OutOfMemory` error. The problem was caused by a hard-coded reference to a JDK in an Ant script. Replacing the reference with a variable fixed the problem.
CR096726	Fixed a problem that caused a NPE when invoking `MBeanHome.deleteMBean()`.
CR097071	Fixed SUID incompatibilities in the System Administration subsystem.
CR099002	Binding is newly implemented for the T3S protocol.

Change Request Number	Description
CR099474	The `getAttributes()` in `RequiredModelMBean` should return a `javax.management.AttributeList`, which should contain `javax.management.Attribute` objects, but in the WebLogic Server 7.0 JMX implementation the `AttributeList` returned by `RequiredModelMBean` contained the value of the MBean attribute directly and not as a `javax.management.Attribute` object, causing the following exception: `java.lang.ClassCastException: java.lang.String at weblogic.management.internal.RemoteMBeanServerImpl.getAttributes (RemoteMBeanServerImpl.java:210) at weblogic.management.internal.RemoteMBeanServerImpl_WebLogic Serverkel.invoke(Unknown Source)` ` at weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.java: 362) at weblogic.rmi.internal.BasicServerRef$1.run(BasicServerRef.java:3 13) at weblogic.security.service.SecurityServiceManager.runAs(SecurityS erviceManager.java:821) at weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServerRe f.java:308) at weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecuteRe quest.java:30) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)End server side stack trace`  `at com.adventnet.beasupport.client.JMXConverter.convertInvocationTa rgetException(JMXConverter.java:744) at com.adventnet.beasupport.client.ProxyMBeanServer.getAttributes(P roxyMBeanServer.java:511) at com.adventnet.beasupport.client.WebLogicClient.getAttributes(Web LogicClient.java:593) at com.adventnet.beasupport.client.WebLogicClient.cmdline(WebLogicC lient.java:1279) at com.adventnet.beasupport.client.WebLogicClient.access$000(WebLog icClient.java:15) at com.adventnet.beasupport.client.WebLogicClient$2.run(WebLogicCli ent.java:1293)`  The `GetAttributes` method now returns an `Attribute` object.

Change Request Number	Description
CR100756	Setting a JMSServer's `Store` property via the `weblogic.Admin` utility:
	`java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic SET -mbean examples:Name=examplesJMSServer,Type=JMSServer -property Store examples:Name=examplesJMSFileStore,Type=JMSStore`
	resulted in the following exception:
	`<Mar 10, 2003 3:32:15 PM EST> <Error> <Management> <141033> <Error importing MBean examples:Name=examplesJMSFileStore,Type=JMSStore to server examplesServer.javax.management.InstanceNotFoundException: examples:Name=examplesJMSFileStore,Type=JMSStore`
	This problem was partially fixed by checking when setting an attribute that is a `WebLogicObjectName` on a configuration MBean. The code checks to see if the bean is a valid one and will throw an `InvalidAttributeValueException`. This is logged on the Adminstration Server and the JMSServer remains usable but does not have the file store set.

Change Request Number	Description
CR101738	A Managed Server failed with the following stack trace:

```
java.rmi.ConnectException: This RJVM has already been shutdown
-1425673232662082784S:WebLogic
ServerDEBUG1:[53101,53101,53102,53102,53101,53102,-1]:WebLogic
Servere1:ADMIN Start server side stack trace:
java.rmi.ConnectException: This RJVM has already been shutdown
-1425673232662082784S:WebLogic
ServerDEBUG1:[53101,53101,53102,53102,53101,53102,-1]:WebLogic
Servere1:ADMIN at
weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java(Compiled
Code)) at
weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java(Compiled
Code)) at
weblogic.rmi.internal.BasicRemoteRef.getOutboundRequest(BasicRem
oteRef.java(Compiled Code)) at
weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.java(
Compiled Code)) at
weblogic.management.internal.AdminMBeanHomeImpl_WebLogic
Servertub.getMBean(Unknown Source) at
weblogic.management.deploy.slave.SlaveDeployer.getTargetVirtualH
osts(SlaveDeployer.java(Compiled Code)) at
weblogic.management.deploy.slave.SlaveDeployer.getCompsFromTask(
SlaveDeployer.java(Compiled Code)) at
weblogic.management.deploy.slave.SlaveDeployer.processRemoveOrDe
activate(SlaveDeployer.java:1240) at
weblogic.management.deploy.slave.SlaveDeployer.commitUpdate(Slav
eDeployer.java:1533) at
weblogic.drs.internal.SlaveCallbackHandler$2.execute(SlaveCallba
ckHandler.java:34) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java(Compile
d Code)) at..
```

Research revealed that the application slave deployer, which runs on the Managed Server, was looking up, caching and using the Administration Server's Admin MBeanHome when it was initialized, that is, when the slave deployer was initialized.

The code was changed so that the Administration Server's Admin MBeanHome is looked up when necessary instead of being cached when the slave deployer is initialized. This resolved the problem.

Change Request Number	Description
CR101856	A cluster contained an Administration Server and two Managed Servers. `Application1.ear` contained EJBs and Web applications; the EJBs were deployed to the cluster and the webApps were deployed to a VirtualHost, which is in turn was targeted to the cluster. `Application1.ear` contained `Application1.war`, which was configured as the DefaultWebApp of the VirtualHost. In addition to this, each cluster node had its own DefaultWebApp.
	Undeploying `Application1.ear` resulted in the following exception:
	``` deactivating application application1 on server2 deactivating application application1 on server1 deactivated application application1 on server2 unpreparing application application1 on server2 deactivated application application1 on server1 unpreparing application application1 on server1 failed application application1 on server2 failed application application1 on server1 Exception caught for task Unprepare application application1 on mycluster,application1: Start server side stack trace: java.lang.reflect.UndeclaredThrowableException: java.lang.reflect.InvocationTargetException: javax.management.ListenerNotFoundException: listener: ServletContext(id=559103557,name=DefaultWebApp,context-path=) ```
	The application could not be redeployed, even if the entire domain was restarted.
	Research revealed that the Application Mbean `unstageTargets()` method used an incorrect indexing for unstaging virtualHosts.
	A code fix resolved the problem.
CR102149	A WebLogic Server instance gave an `OutOfMemoryError` when receiving 21,000 persisted messages synchronously in a transacted session on Solaris.
	The garbage collector was slow at reclaiming long lists that are freed. Breaking up the linked lists fixed the problem.
CR105736	Some load-balancing applications such as Cisco Loadbalancer became unable to use session stickiness because of a change to the SessionID format in 7.0.
	A new server startup flag, `-Dweblogic.servlet.useExtendedSessionFormat=true`, retains the information that the load-balancing application needs for session stickiness.

Change Request Number	Description
CR105777	Exception propagation was not working properly because `RemoteException` was not being treated as a valuetype for rapid caching.
	Caching only real valuetypes fixed the problem.
CR105831	An exploded Web application was allowed to load classes from the docroot.
	This inappropriate classloading behavior has been disallowed.
CR106198	On `GroupReaderMBean`, `isMember` was failing when member name contained a comma.
	A code change resolved this problem so that member names containing commas are now accepted.

Change Request Number	Description
CR106687	Links on the Examples server point to local files, which does not work if users are accessing them from remote browsers.
	The problem was fixed by adding virtual-mapping to `weblogic.xml` of examplesWebApp so that remote users may browse example documentation.
	To activate, properly configure `virtual-mapping` in `weblogic.xml`, replacing the `<local-path>` value of `<virtual-directory-mapping>` with your samples installation location (for example, `d:/bea/weblogic/samples/server/src`).
CR107043	When a t3 connection was established with WebLogic Server with `weblogic.jar` in the bootclasspath, the following exception occurred:
| | ```
java.lang.NullPointerException at
weblogic.rmi.internal.StubGenerator.getStubClass(StubGenerator.j
ava:632) at
weblogic.rmi.internal.StubGenerator.generateStub(StubGenerator.j
ava:656) at
weblogic.rmi.internal.StubGenerator.generateStub(StubGenerator.j
ava:651) at
weblogic.rmi.extensions.StubFactory.getStub(StubFactory.java:59)
at
weblogic.jndi.WLInitialContextFactoryDelegate.newRootNamingNodeS
tub(WLInitialContextFactoryDelegate.java:489) at
weblogic.jndi.WLInitialContextFactoryDelegate.newRemoteContext(W
LInitialContextFactoryDelegate.java:452) at
weblogic.jndi.WLInitialContextFactoryDelegate.newContext(WLIniti
alContextFactoryDelegate.java:372) at
weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext(
WLInitialContextFactoryDelegate.java:339) at
weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext(
WLInitialContextFactoryDelegate.java:221) at
weblogic.jndi.WLInitialContextFactory.getInitialContext(WLInitia
lContextFactory.java:149) at
javax.naming.spi.NamingManager.getInitialContext(NamingManager.j
ava:671) at
javax.naming.InitialContext.getDefaultInitCtx(InitialContext.jav
a:242) at
javax.naming.InitialContext.init(InitialContext.java:218)
``` |
| | A code change that specifies using an augmentable classloader to generate stubs fixed the problem. |

Tools

Change Request Number	Description
CR074714	In the WebLogic Builder WebLogic Settings panel, when a RAR's "Capacity increment" was larger than the difference between "Maximum capacity" and "Initial capacity", WebLogic Builder failed to load the RAR, and printed a SAXProcessorException beginning: `weblogic.xml.process.: capacity-increment should be less than or equal to ( max_capacity-initial_capacity)` A code change resolved the problem.
CR072039	Relations listed in the EJB Relations panel are now sorted alphabetically by name.
CR073975	Method name and transaction attributes are now required information for creating a new method.
CR074198	Methods from the bean class are no longer listed in the Method name selection list in the Add method dialog. Home, local home, remote interface, and local interface methods are still listed.
CR074591	The deployment dialog no longer needs to be refreshed before it displays accurately the status of a failed deployment.
CR075935	In the Advanced panel of an entity bean, the selection list was not displaying all of the read-only entity beans that can be invalidated when the container-managed persistence entity bean has been modified. The list now displays all such beans.
CR078807	When you open a project without compiling the classes referenced in the deployment descriptors, Builder used to throw a spurious AssertionError. It now warns the user that the classes cannot be found.
CR081472	WebLogic Builder no longer incorrectly allows JNDI names to be assigned to beans that have no local interface.
CR082276	WebLogic Builder's validation utility detects updated class files in exploded modules, but not in archived modules. Builder detects updated class files in both archived and exploded modules when you close and re-open the project
CR085188	If you remove all CMP fields from a project, Builder may corrupt `weblogic-rdbms-cmp-jar.xml`. A warning appears if you try to remove all CMP fields from a project.
CR085270	It is now possible to abort or cancel creation of a relationship in Builder's Add Relation, Add Finder, and Add CMP wizards.

Change Request Number	Description
CR088145	Builder now displays a warning when invalid CMP nodes are activated.
CR091665	Before this release, adding a row in a bean's CMP Fields node and then deleting it in the table view would cause an AssertionError to be thrown. This problem, which was caused by an incompletely defined property in Builder's user interface, has been fixed.
CR091720	An advanced panel now exists for BMP beans, similar to the panel for CMP beans.
CR093246	A new deployment descriptor element called `<allow-remove-during-transaction>` was added to the `weblogic-ejb-jar.xml` deployment descriptor. This tag was added for stateful session beans and is the child of the `<stateful-session-descriptor>` tag. However, the tag was not surfaced in WebLogic Builder.
	This problem was resolved by adding the 'Allow remove during transaction' check box to the Advanced Pane of a Session in WebLogic Builder.
CR093579	Using `weblogic.Deployer` to deploy an application to a cluster with a user in the Deployers group deployed the application, but also caused the following exception:
	`weblogic.management.NoAccessRuntimeException: Access not allowed for subject: principals=[velvet, Deployers], on ResourceType: Cluster Action: execute, Target: addDeployment`
	`        at weblogic.management.internal.Helper$IsAccessAllowedPrivilegeAction.run(Helper.java:2034)`
	Adding ACLs for cluster deployment fixed the problem.
CR093658	Calling `getLogFileName` on the `VirtualHostMBean` was returning an incorrect path.
CR093833	Redeploying to a Managed Server using `weblogic.Deployer` without specifying the server was deploying the application to the Administration Server.
	The problem has been resolved by a change to the `weblogic.Deployer` utility.

Change Request Number	Description
CR094825	`weblogic.Admin` was not handling invalid URL arguments properly. For instance: `java weblogic.Admin https://hostname:8888` when there is no server running there should result in a "Failed to connect" error message, but sometimes resulted in the command exiting without explanation, or hanging. The problem was fixed. An exception that was being thrown by `javax.net.ssl.SSLSocketImpl.close` is now caught, and an appropriate error message is displayed.
CR095915	Creating a relation with the same name as another relation in the same project is no longer allowed in WebLogic Builder.
CR096615	The cache table in the Tuning panel of an entity bean is now disabled if no cache references are available. Clicking on elements of a cache table when no cache references were available was causing an error.
CR096719	WebLogic Builder did not expose pool settings for entity beans. This problem was resolved by adding a Pool Panel for entity beans to the Tuning sub-node.
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.
CR098134	`weblogic.Admin` prompted for a password, and then threw an exception before you could type it. A code change resolved this problem.
CR098709 Tools	While performing a SET operation, the client was constructing `WebLogicObjectname` corresponding to the MBean. However, other members such as parent, were not set correctly. This made the `WebLogicObjectname` inconsistent. Sanitizing all attributes whose values belong to `WebLogicObjectname` fixed the problem.

Change Request Number	Description
CR098985	The `setWebLogic ServerEnv.sh` script sometimes failed to source the `extEnv.sh` script. The problem was a relative address of the `extEnv.sh` script that did not work in all system shells. Replacing the reference with a universally readable relative address fixed the problem.
CR099260	You can now rename existing relations in the Add relations wizard. The name field of an existing relation was previously read-only.
CR099865	Previously, resetting default transaction attributes in a bean was causing transaction attributes in other beans in the project to reset also. Multiple methods containing multiple transaction attributes were sometimes redefined as a single method with a single transaction attribute. This behavior no longer occurs.
CR099911	Previously, some beans with bean-managed persistence were timing out at periods inconsistent with the transaction timeout settings in their deployment descriptor files. BMP beans now behave as CMP beans do, in accordance with the deployment descriptors.
CR099913	In previous versions of WebLogic Builder, some controls were inappropriately available. For instance, you could fill out the name value for an Oracle sequence even if automatic key generation was not enabled for the application. An effort has been made to disable and enable controls appropriately throughout the WebLogic Builder interface.
CR099913	The automatic key generation frame in WebLogic Builder did not enable/disable all fields consistently. For example, if you opened a JAR that did not have an `<automatic-key-generation>` tag and loaded the Automatic Key Generation tab, the check box would be unchecked, but you could still enter a value. This problem was resolved by modifying the code so that the fields are uniformly disabled.
CR099917	The setting for caching between transactions was not being persisted, so that you could go to a bean's Tuning panel and select a concurrency strategy and check "Cache between transactions," and when you re-opened the project the setting had reverted to "False." This setting is now persisted.
CR100288	Updates to finder methods were sometimes not being persisted. The autocommit setting was to blame, and it has been fixed.

Change Request Number	Description
CR100756	The Server was sometimes unusable until restart after a bad Type setting had been used when setting `JMSFileStore` with `weblogic.Admin`. A bad Type setting now results in a `InvalidAttributeValueException` being logged on the Admin server.
CR103174	Debug flags in the command-line to start a server were persisting to the domain configuration file, so that a server started in debugging mode remained in debugging mode until the debug flags were taken out of `config.xml`. Following a code fix, command-line flags no longer write to `config.xml`.
CR105436	`java weblogic.marathon.ddinit.WebInit stageDir` failed to generate servlet components in `weblogic.xml` and `web.xml`. A regression in the codeline cause the problem. This was fixed and the problem was resolved.

Web Services

Change Request Number	Description
CR106013	In client code with `weblogic.jar` in the classpath `java.net.HttpURLConnection.getResponseCode` returned 200, while running in the container `java.net.HttpURLConnection.getResponseCode` for the same url returned 404. The cause of the 404 message was that, using `HttpURLConnection` from within WebLogic Server, the EJB code received an instance of `weblogic.net.http.HttpURLConnection`. `setRequestProperty()` ignored any headers with empty value strings. SOAPAction header was required by the Apache server hosting `http://www.xmethods.net/interfaces/query`.

You can specify which protocol Handler you want to use, which allows you to use the Sun implementation in your code. The following simple servlet illustrates this:

```
package test;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.net.*;
public class ProtocolHandlerTest extends HttpServlet {
    public void service(HttpServletRequest request,
HttpServletResponse response) { try { URL testUrl_1 = new
URL(null, "http://www.yahoo.com:80/";, new
sun.net.www.protocol.http.Handler());
            URLConnection huc_1 = testUrl_1.openConnection();
            URL testUrl_2 = new URL(null,
"http://www.yahoo.com:80/";);
            URLConnection huc_2 = testUrl_2.openConnection();
            System.out.println("huc_1 =
"+huc_1.getClass().getName());
            System.out.println("huc_2 =
"+huc_2.getClass().getName());
        } catch (java.net.MalformedURLException mue) {
        } catch (java.io.IOException ioe) { } }}
```

Change Request Number	Description
CR078871	The WebLogic Web Services runtime environment, when generating the WSDL for a deployed Web Service, now correctly maps any exceptions thrown by the EJB method that implements an operation to a `<soap:fault>` element in the WSDL. Previously, these Web Service-specific exceptions were being incorrectly mapped to `<soap:body>` elements.

Change Request Number	Description
CR079631	The `clientgen` Ant task now correctly accepts a `<classpath>` child element in the `build.xml` file that contains the call to `clientgen`.
CR079776	The Javadoc for the `weblogic.uddi.client.service.Inquiry.findService` method of the UDDI Client API has been updated to specify that, in addition to the service name, you must also include the business key when using the method.
CR082779	The implementation of the `weblogic.webservice.client.WebLogic ServerSLAdapter` interface no longer fails to propagate user certificates to the SSL connections it opens. Previously, it was not possible to perform authentication on Web Service invocations using client-side certificates over SSL.
CR083861	The WebLogic Web Services runtime no longer alters the `javax.xml.rpc.soap.SOAPFaultException` exception when thrown from a stateless session EJB that implements a WebLogic Web Service. Previously, the exception would be incorrectly wrapped in an `EJBException`.
CR084960	The Web Services runtime now correctly propagates the HTTP 500 "Internal Server Error" exception to a client if the Web service being invoked throws an exception. The actual exception is correctly contained in the SOAP Fault.
CR087303	When processing the following WSDL file, the clientgen ant task created invalid classes: `wsdl:message name="UploadCBProfileRequest"> <wsdl:part name="body" element="tns:TRAMSDATA"/> <wsdl:part name="authHeader" element="tns:AuthenticationHeader"/> </wsdl:message> <wsdl:portType name="RelMgrPortType"> <wsdl:operation name="uploadCBProfile"> <wsdl:input message="tns:UploadCBProfileRequest"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="RelMgrSOAPBinding" type="tns:RelMgrPortType"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http";/> <wsdl:operation name="uploadCBProfile"> <soap:operation soapAction="" style="document"/> <wsdl:input> <soap:body parts="body" use="literal"/> <soap:header wsdl:required="true" message="tns:UploadCBProfileRequest" part="authHeader" use="literal" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/";> </soap:header> </wsdl:input> </wsdl:operation> </wsdl:binding>`    The resultant generated task contained duplicate `uploadCBProfile()` arguments. Research revealed that the code was adding header parts to message even if that part was already there. A code fix resolved the problem.

Change Request Number	Description
CR087529	In WebLogic Server 6.1SP03, the client for the sample Web service at `samples\examples\webservices\rpc\weatherEJB` was built by `wsgen` and deployed using only the client.jar (no `weblogic.jar`). When the non-SSL web service was invoked by the Java client, the following exception was thrown: `<< Missing class files in servicegen client.jar. Getting Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/net/http/HttpsURLConnection at weblogic.soap.http.SoapContext.lookup(SoapContext.java:87) at javax.naming.InitialContext.lookup(InitialContext.java:350) at com.verizon.client.client.main(client.java:72) >>`  Analysis revealed that the `weblogic.net.http.HttpsURLConnection` class is necessary for minimal out-of-the-box functionality and should be included in the `client.jar`.  The problem was resolved by adding `weblogic.net.http.HttpsURLConnection` to `client.txt`.
CR090882	You can now successfully use the UDDIExplorer to publish a Web Service to a private UDDI registry when using the JRockit JVM with WebLogic Server. Previously, publishing a service to the private registry would return the UDDIExplorer error "E_fatalError(10500): a serious technical error has occurred while processing the request." The same publish operation has always worked correctly with the default Sun JVM.
CR092549	The Web Service Ant tasks that generate the non-built-in data type components (such as the serialization class, Java class, and so on) now correctly convert the unsupported XML Schema data type xsd:union to the default Java class for unsupported XML Schema types: SOAPElement. Previously the Ant tasks would fail with an error.
CR095044	The WebLogic Web Services runtime now wraps the details of exceptions it gets from WebLogic Server in CDATA, correcting a previous problem in which the details of a SOAP fault might contain invalid XML. An example of invalid XML is a WebLogic Server message such as `<no stracktrace available>` is placed within `<>` resulting in the invalid `<<no stacktrace available>>`.

Change Request Number	Description
CR095109	Unless verbose mode is explicitly enabled, the following warning is no longer outputted to the command window from which you started WebLogic Server when deploying a Web Service that has a handler-implemented operation: WARNINIG: Unable to find a javaType for the xmlType:*datatype*. Make sure that you have registered this xml type in the type mapping Using SOAPElement instead. The Web Services runtime is checking for typemapping information which is not needed in this case, so the warning is confusing and misleading.
CR095561	The `servicegen` Ant task no longer returns an error when used with the `typeMappingFile` attribute, and the specified `types.xml` file contains a class name for a non-built-in Java data type that does not follow the JAX-RPC naming conventions. If you are creating your own serialization class, non-built-in Java data types, and so on, you can name the Java data type anything you want.
CR095741	The WebLogic Web Services runtime now correctly handles the time part of `xsd:dateTime` values. Previously, due to incorrectly handling of timezone offsets, the time would change depending on the timezone.
CR096906	The WebLogic Web Services runtime now correctly handles `xsd:dateTime` data types. Previously, the "+" character was omitted from the time zone offset part of the dateTime value when the timezone was positive.
CR103937	On abort of an SSL connection, the CLOSE_WAIT might remain for some time, keeping the socket open. A code change fixed the socket leak.
CR104719	Implementing a Web service with the following signature: `public void echoDom(Document doc)` resulted in a compile error. This problem was resolved by disallowing `void` to be added to type-mapping for document style at build or runtime, preventing the writing of `void` part for document style, and avoiding using the wrapper element in the Document serializer for document style.
CR104783	`clientgen` was not creating serializer and deserializer classes for complex types. `clientgen` now handles complex types that contain single references to model groups.

Change Request Number	Description
CR106392	Running with invocation-style="one-way" results in the WSDL containing an <output> in the <binding> section for that operation as can be seen from the snippet below.

```
<portType name="MyServicePort">
<operation name="sayNothing">
 <input message="tns:sayNothing" />
 </operation>
 </portType>
 <binding name="MyServicePortSoapBinding"
type="tns:MyServicePort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"; />
 <operation name="sayNothing">
  <soap:operation soapAction="" style="document" />
 <input>
  <soap:body use="literal" namespace="http://tempuri.org/"; />
  </input>
 <output>
  <soap:body use="literal" namespace="http://tempuri.org/"; />
  </output>
  </operation>
  </binding>
```

This behavior was exhibited in both "rpc" and "document" style web services. Running with Microsoft's WSDL.EXE against this WSDL resulted in an unsuccessful validation and consequently in a failure to generate the .Net client proxy or stubs.

A code change stopped the generation of output messages in one-way bindings, resolving the problem.

Change Request Number	Description
CR107254	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.
CR108332	When using a thin client (`wlclient.jar`), customers may have encountered a marshalling exception that indicated an IIOP stream corruption. This problem occurred when IIOP or T3 was specified as the protocol in multiple minor versions of J2SE 1.4, but not when `weblogic.jar` was in the client classpath.
	This problem occurred because of issues with the encoding and decoding of custom marshaled valuetypes and in the way WebLogic Server kept track of indirections when marshaling valuetypes.
	These issues have been resolved.

WebLogic Tuxedo Connector

Change Request Number	Description
CR074963	When creating large Field Manipulation Language (FML) tables (on the order of 4K entries), `mkfldclass[32]` 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. For more information, see Using the DynRdHdr Property for mkfldclass32 Class at http://e-docs.bea.com/WebLogic Server/docs70/wtc_admin/XML_FML.html.

Change Request Number	Description
CR089100	`viewj` and `viewj32` methods can create huge java classes (exceeding 65535 byte codes in the constructor method) when compiling view files that contain a large number of fields.
	Analysis of the problem revealed that majority of the code in the constructor involved assigning null values to fields and array elements and could be removed.
	This problem was resolved by providing a code fix that suppresses redundant code when compiling view files.
CR092860	Using WTC to access Tuxedo 8 Corba objects resulted ia n `JavaIDL Reader` thread leak. Running a simple example application resulted in a build-up of `JavaIDL Reader` threads in the JVM running WebLogic Server. The threads were not destroyed until the server was shut down.
	Analysis revealed the `finalize()` was not called at the correct time. A code fix to add a call to `ORB.destroy()` was implemented to resolve the problem.

XML

Change Request Number	Description
CR087201	The built-in XML parser (based on Apache's Xerces) no longer throws a java.lang.ArrayIndexOutOfBoundsException exception when parsing an XML file that contains Japanese characters.
CR090137	The built-in XML transformer (based on Apache's Xalan) now correctly adds a closing </META> tag when transforming an XML file into an HTML file using an XSL stylesheet. Previously, the closing </META> tag was not generated in the output HTML file, causing the XML to be invalid, although the HTML file rendered correctly in a browser.

Change Request Number	Description
CR095073	Previously, certain types of XML files caused a decay in the response times of parsing them as the iterations increased. This is no longer true.
CR100068	There was a conflict using JSTL tags in JSP with Japanese characters under these conditions:

CR100068 (continued):

1. Page encoding of JSP is defined by Shift_JIS. <%@ page pageEncoding="Shift_JIS" %>

2. Using multibyte characters (Japanese) in JSP.

3. Using JSTL tags.

The following errors occurred when executing the JSP:

```
java.io.IOException: javax.servlet.jsp.JspException: The taglib
validator rejected the page: "org.xml.sax.SAXParseException: An
invalid XML character (Unicode: 0x82) was found in the CDATA
section., " at
weblogic.servlet.jsp.Jsp2Java.outputs(Jsp2Java.java:124) at
weblogic.utils.compiler.CodeGenerator.generate(CodeGenerator.jav
a:258) at
weblogic.servlet.jsp.JspStub.compilePage(JspStub.java:353) at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:211) at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:164) at
weblogic.servlet.internal.ServletStubImpl.getServlet(ServletStub
Impl.java:517) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:351) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:445) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletS
tubImpl.java:306) at
weblogic.servlet.internal.WebAppServletContext$ServletInvocation
Action.run(WebAppServletContext.java:5445) at
weblogic.security.service.SecurityServiceManager.runAs(SecurityS
erviceManager.java:780) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(Web
AppServletContext.java:3105) at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequ
estImpl.java:2588) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)
```

Research revealed that the encoding for makeXMLStream was incorrect. Changing the encoding to UTF-8 resolved the problem.

WebLogic Server 7.0 Service Pack 2 Solutions

The following sections describe problems that were resolved for the release of WebLogic Server 7.0 Service Pack 2.

- "Administration Console" on page 4-261
- "Classloader" on page 4-262
- "Cluster" on page 4-264
- "Connectors" on page 4-266
- "Core Server" on page 4-267
- "Deploy" on page 4-270
- "EJB" on page 4-271
- "Examples" on page 4-275
- "Installer" on page 4-275
- "Internationalization" on page 4-275
- "jCOM" on page 4-276
- "JDBC" on page 4-276
- "jDriver" on page 4-278
- "JMS" on page 4-278
- "JNDI" on page 4-281
- "JTA" on page 4-281
- "JTS" on page 4-282
- "Miscellaneous" on page 4-282
- "Node Manager" on page 4-284
- "Plug-ins" on page 4-285
- "RMI/RMI-IIOP" on page 4-286
- "Security" on page 4-288

Administration Console

Change Request Number	Description
CR074370	Making changes to an EJB descriptor value via the Deployment Descriptor Editor provoked a `javax.management.AttributeNotFoundException`. This problem has been resolved.
CR074653	The Administration 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 or deployment can fail.
CR074677	When you upgrade a domain created under version 6.x of WebLogic Server, the user "system" now explicitly grants access to the Administration Console.
CR074933	Translated online help was not visible; now it is visible.
CR079805	Web applications that are targeted to the source server now show up as targeted to the cloned server.
CR081366	Machine deletion no longer fails when one or more servers are configured on it.
CR081445	There are no longer empty monitoring pages for EJBs with the Netscape 4.7 browser.
CR082345	In WebLogic 7.0.0.0, the Console threw an NPE when JDBC driver was not in $CLASSPATH. This problem has been fixed in this release.
CR082942	Changing a user password and then clicking Continue no longer throws a `NullPointerException`.
CR083242	Users can now reboot a server when a Realm Adapter ATN provider is added to a 7.x security realm.
CR083330	Users can now target a multipool to a cluster in the Console.

Change Request Number	Description
CR083465	Clicking on Server --> Services --> Bridge --> Monitor all Messaging Bridge Runtimes no longer causes a 404 error.
CR083530	There is now a Console configuration option to for specifying that the log be timestamped in Greenwich Mean Time (GMT).
CR084558	Attempting to edit the EJB Deployment descriptors of Pipeline EJB shipped with the sample WebLogic Portal Server no longer fails with MBean exceptions, such as javax.management.AttribtuteNotFound.
CR084856	"Security" -> "Users" -> "Unlock Users" is no longer garbled in a Japanese environment.
CR085101	EJB methods can now define policies correctly via the Console.
CR086107	The console now uses the global Admin/Operator/Monitor/Deployer roles.
CR087802	Editing EJB deployment descriptors no longer provokes a `NullPointerException`.
CR089044	Browser warning pages no longer display in the Administration Console when a browser is not supported.
CR092102	An Administration Server no longer throws an `OutOfMemory` exception when the Administration Console is set to monitor thread pools.

Classloader

Change Request Number	Description
CR056911	Two different `.war` files within an `.ear` file each instantiated their own instance of a singleton, resulting in incorrect results when the application was run. This problem has been fixed.
CR081377	Loading classes via the context class loader at deployment time no longer fails and causes `ClassNotFound` exceptions.

Change Request Number	Description
CR082466	Multiple entries were being added to the ClassFinder. As a result, `ClassLoader.getResources` returned an Enumeration with duplicate entries. The problem has been fixed.
CR083752	A client class can now successfully create an InitialContext when the java command line specifies `weblogic.jar` using the `-Xbootclasspath` argument.

Cluster

Change Request Number	Description
CR043366	Under WebLogic Server 6.1, running on any platform, when the Cluster Address was specified as a list of comma separated IP addresses, for example:
	`IPaddress1, IPaddress2`
	or
	`IPaddress1:7011, IPaddress2:7011`
	and error message was issued when the Managed Servers in the cluster started up:
	`<main> <system> <> <000101> <Cannot resolve ClusterAddress: IPaddress1:7011, IPaddress2:7011`
	The cluster started correctly. The comma delimited could not be resolved.
	A code change was implemented to support ClusterAddress in format of a list of comma separated IP addresses, when the domain is set to Development mode. This format is not supported for projection mode operation. EJB handles may not be managed properly, resulting in problems with failover when this format is used.
	In Production mode, the cluster address must be specified as a DNS name that maps to the Managed Servers in the cluster.
CR074236	Timing issues related to creation of the cluster-wide JNDI tree and deployment of MDBs resulted in a NameNotFoundException. during startup. The `ejbCreate` method of the MDB called a service on an EJB that was pinned to a single node in the cluster. For reasons related to application requirements, this node is started first. When a secondary node started, and the MDBs deployed with an `initial-beans-in-pool` setting greater than 0 the beans threw the NameNotFoundException. This occurred because the home interface for the pinned EJB had not been bound to the JNDI tree yet. When the MDBs were deployed with an initial pool setting of 0 the node started correctly and the beans work properly when the first message is queued because JNDI has had a chance to catch up.
	This problem was solved by implementing a `MemberWarmupTimeoutSeconds` attribute in `ClusterMBean`. The default value is 0 which means no cluster warmup. By default, the cluster startup sequence remains unchanged. If a value > 0 is set, cluster member's wait for that period of time to synchronize with other members during startup. All MDBs are deployed after the warmup time.

Change Request Number	Description
CR078455	A six node cluster hosting a Web application ran for seven hours, then hung repeatedly. The configuration was: F5/Big IP Load Balancer directing requests to cluster of three servers running WebLogic Server 6.1 SP2, IIS and ISAPI plug-in, in turn proxying to a cluster of six WebLogic Server 6.1 SP2 servers. Host machines were Solaris 8 dual CPU servers running JDK 1.3.1.
	The plug-in directed requests to a server instance other than primary or secondary, and WebLogic Server looked up and made calls on remote `ROIDImpl`.
	The problem was solved by a fix that fabricates a local stub for `ROIDLookup` and Replication Manager RMI objects, thus avoiding JNDI lookups for the stub. In addition, calls to `ROIDLookup` now use the replication threads to ensure execution even when the default execute threads are unavailable, avoiding a possible hang.
CR078677	A Web application was deployed on a cluster. `HttpClusterServlet` did not fail-over during the graceful shutdown of a Managed Server under WebLogic Server6.1 Sp2. The Web application was undeployed and could not find a match for the context when request was received during the shutdown sequence. Instead of transparent failover, a page not found error was displayed in the browser. This problem did not occur in the case of a a server crash, only during a graceful shutdown.
	This problem was fixed by implementing a check for server state, and logic to return a `SERVICE_UNAVAILABLE (503)` when context is null, allowing plug-ins to fail-over instead of returning `404 (NOT FOUND)` to the client.
CR080153	On an NT multi-homed system, an Administration Server in Managed Server discovery mode did not restart if the URL host was the same as the host name, and the host name resolved to multiple IP addresses. When the host name resolved to multiple addresses, the Administration Server cannot reliably determine which Managed Servers it was managing, prior to failure, hence restart of the Administration Server did not complete.
	A check was added to `Admin.java` to determine, when user specifies a dotted IP address for the Administration Server when starting a Managed Server, if the address is multihomed - in which case this message is displayed: `'The admin server url "http://qa700:9001"; cannot be used with managed-server-discovery-mode because the hostname resolves to multiple ip addresses. Please turn off discovery mode or set just one ip address for this host.'`
CR080976	Fixed a problem in which shutting down an Administration Server, with a timeout option, resulted in this error message: `Expected RemoteException, RuntimeException, or Error but received: 'class weblogic.rjvm.PeerGoneException.'`
	Error has been corrected. Now, when shutting down with a timeout option, the following message is output: The shutdown sequence has been initiated.

Change Request Number	Description
CR082443	The debug flag for clusters could not be turned off, resulting in large log files. The problem occurred because, in `weblogic.cluster.ClusterDebug.java`, the flag controlling whether or not debug messages are logged was set to true, and could not be changed. The problem was solved by providing the ability to configure the flag using a tag in `config.xml`: `<ServerDebug DebugCluster="true" Name="myserver"/>`
CR083532	Null Pointer Exception at `ReplicaAwareRemoteRef.getCurrentReplica` resulted in termination of shutdown process before the shutdown was complete. This error was corrected.

Connectors

Change Request Number	Description
CR080250	Deploying a RAR without a `weblogic-ra.xml` descriptor file no longer provokes a `NullPointerException`.
CR082608	Fixed a problem in which the JCA Connection Factory would try to obtain a new connection instead of using the available connections.
CR086251	During shutdown of WebLogic Server, Messaging Bridge Adapter undeployment no longer throws a `NullPointerException`.
CR089279	WebLogic Server no longer persists deprecated default `weblogic-ra.xml` elements.

Core Server

Change Request Number	Description
CR032447	Emergency messages are no longer issued during a deliberate server shutdown.
CR058358	The `alt-dd` deployment descriptor element is now handled correctly for EJBs and Web applications.
CR062756	Socket reads are no longer blocked, causing a server to hang with JDK version 1.3.1.
CR068646	`weblogic.admin` no longer succeeds at HTTP tunneling when tunneling is disabled in the Server Mbean.
CR070429	You can now re-boot with the last known good COMMO configuration, by booting the server with the `-Dweblogic.safeCommoBoot=true` flag.
CR071796	A server launched from a non-default directory can now see its applications.
CR074265	Using MAX_BACKOFF_BETWEEN_FAILURES on weblogic.t3.srvr.ListenThread, users can now configure the time period in which the server listen thread will send a "Failed to listen on port" message if it is still trying to accept a socket connection. Please refer to the weblogic.t3.srvr.ListenThread javadoc getMaxBackoffBetweenFailures() and setMaxBackoffBetweenFailures() for more information.
CR077831	When the AppletArchiver utility is used to generate a client jar file and the applet in question uses the `javax.swing.JApplet` class in its code, an error is no longer thrown.
CR079547	`InvalidClientIDException` and `NameAlreadyBound` exceptions no longer occur when creating Durable subscribers to JMS servers in a clustered environment.
CR080744	Various multiplexor (muxer) problems were resolved.
CR081176	The `getState` and `getStateVal` properties in the `ServerLifeCycleRuntimeBean` now behave similarly.
CR082026	`JavaSocketMuxer` no longer creates a new thread every time WebLogic Server registers a socket on client.
CR082348	Shutting down `wlidomain` via WebLogic Server console no longer causes a `java.lang.ThreadDeath`.

Change Request Number	Description
CR082533	In earlier versions/service packs,(6.1SP3) some debug messages from ChangeAwareClassLoader were inappropriately written to the stdout. This bug has been fixed in this release.
CR083654	The performance of WebLogic Server 6.1SP3 is somewhat affected during continuous requests from Microsoft WAS Tool. Performance is improved with fix for JSPs and servlets that use a multibyte charset as content-type in the response.
CR084977	The "root" identity no longer retains control of `.wlnotdelete` after server startup.
CR085544	Invoking `weblogic.Server.stop()` from NT Service to shut down a WebLogic Server instance no longer throws a `NullPointerException`.
CR085659	The synchronized code in the RJVM no longer restricts EJB failover during machine disconnection.
CR085669	On HPUX, SAP JCO no longer causes Java Virtual Machine (JVM) crashes.
CR086108	WebLogic Server 6.1SP3 has a performance regression in ListenThread as it consumes more CPU time when compared with WebLogic Server 7.0 SP2. The server throttling was causing performance degradation by caching information about the active sockets. This problem has been fixed.
CR086425, CR84172	Large Message Transmission Units (MTUs) no longer cause performance degradation.
CR086552	`java.io.OptionalDataException` is no longer thrown (in WebLogic Server 7.0SP1) during Java serialization.
CR086758	In an application with a web tier and an EJB tier, the web tier communicating with the EJB tier no longer hangs and throws exception similar to the one shown below when the EJB tier closes the ConnectionManager. `<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(ConnectionManager .java:1348) at ...`

Change Request Number	Description
CR087180	A connection from a Managed Server to an Administration Server no longer terminates prematurely before shutdown of Managed Server is complete. An exception similar to the following used to be thrown in 7.0SP01. `<Oct 1, 2002 3:22:42 PM PDT> <Warning> <rmi> <080006> <Failed to associate the transaction context with the response while marshalling an exception: java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@7d37fa - id: '2847122412264039151S:172.17.24.65:[7001,7001,7002,7002,7001,7002,-1]:mydomain:myserver' connect time: 'Tue Oct 01 15:18:40 PDT 2002'' has already been shut down java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@7d37fa - id: '2847122412264039151S:172.17.24.65:[7001,7001,7002,7002,7001,7002,-1]:mydomain:myserver' connect time: 'Tue Oct 01 15:18:40 PDT 2002'' has already been shut down at weblogic.rjvm.ConnectionManager.getOutputStream(ConnectionManager.java:1564) at`
CR087254	An IIOP performance regression was resolved.
CR088056	Build dates are now embedded in multiplexor (muxer) libraries.
CR088415	Setting `ulimit -n` to `unlimited` no longer causes server crashes on AIX.
CR089035	Accessing a simple stateless session bean in a cluster no longer provokes `IllegalArgumentException: port out of range`.
CR089144	Deployment of EJBs with a custom `CallRouter` no longer fails with an `IllegalArgumentException`.
CR092183	The `PosixMuxer` no longer throws `Interrupted System Call` exceptions.
CR092375	Manually doing a lookup and then persisting and caching the handle did not work when the methods were conversational. This is fixed with changes to serialization and deserialization of EJBHandle object.

Deploy

Change Request Number	Description
CR049340	When installing an application through the console or dropping an application in the applications directory in development mode, the components now deploy in the order defined in `applications.xml`.
CR080216	Hot-deploying or manually deploying a Web application intermittently no longer results in a `NoSuchObjectException: Bean is already undeployed` exception.
CR080537, CR084907	Fixed a problem in which a startup class marked to be deployed before applications—because the `LoadBeforeAppDeployments` was set to `True`—nonetheless were loaded after applications were deployed, on server startup.
CR080929	`weblogic.refresh` now works for files whose names are specified using wildcard characters.
CR081311	A `SaxParseException` on the DTD now has a severity of `Warn`, rather than `Info`.
CR082263	There is no longer a cross-platform path incompatibility in weblogic.deploy. weblogic.deploy now successfully updates the webapp (`.war` file) from a Windows 2000 machine to a cluster running on Solaris.
CR083179	When `web.xml` and `weblogic.xml` are out of sync with respect to the resource-ref and resource-description elements, the error message now names the ResourceReference in question, rather than giving generic information.
CR083652	Applications are now deployed once at startup, instead of twice.
CR085762	An application targeted to a Managed Server now correctly deploys even if the Managed Server is not running.
CR085979	`MBeanHome.getAllMBeans()` no longer throws a `java.lang.IllegalArgumentException` when a custom MBean is defined.

Change Request Number	Description
CR087830	Applications targeted to Managed Server(s), are now correctly reported as deployed even after an Administration Server shutdown and restart from the Administration Console.
CR089098	This problem has been resolved: A Web application targeted to a Virtual Host while the Virtual host is targeted to a Managed Server. Changes are made to the JSP in the Web application. An undeploy followed by a redeploy fails, in that changes made in the Web application cannot subsequently be seen in the browser when a new request is made.

EJB

Change Request Number	Description
CR062512	Fixed a small performance regression introduced in 7.0 with Stateful Session Beans, with tx-attributes of `Required` and `NotSupported`.
CR060965	Spaces are no longer required around an equals sign ('=') in an EJB QL query.

Change Request Number	Description
CR063275	WebLogic Server EJB did not provide the ability to write a finder method on a byte-cmp 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 cmp-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: `ejbc:` `[java]` `[java] ERROR: Error from ejbc: Error while reading 'META-INF/weblogic-cmp-rdbms-jar.xml'. The error was:` `[java]` `[java] invalid query: In EJB containerManaged, for a query defined in the ejb-jar.xml file with a method signature, findFk(byte[]), we failed to find a corresponding method in the remote home interface, local home interface, or bean class that matches this signature. Note that class parameters such as java.lang.String must be fully qualified, thus 'String' would not match 'java.lang.String'.`  `[java]`  WebLogic Server EJB now supports finder methods on a `byte-cmp` field.
CR063502	Automatic primary key generation is now supported for `java.lang.Long`.
CR076272	Error reporting is now improved for hot-deployed EJBs.
CR079616	A subquery without a `WHERE` clause no longer generates incorrect SQL.
CR079745	`ejbc` now performs deployment correctly when another `.xml` file is included in `weblogic-ejb-jar.xml`.
CR080030	An `is-modified-method-name` tag in `weblogic-ejb-jar.xml` no longer results in an `java.lang.NoSuchMethodException`, when updating an EJB 1.1 bean to EJB 2.0.
CR080569	EJBCacheRuntime statistics no longer report negative numbers for values.
CR080985	If an in-memory replication license is not detected, the EJB container now handles the situation gracefully, rather than throwing an error.
CR081735	The `dispatchPolicy` option is now a public option for both `ejbc` and `rmic`.
CR081753	Repeated attempts to deploy and undeploy a bean no longer cause the server to run out of memory.

Change Request Number	Description
CR081807	When `home-is-clusterable` was set to `False` the server no longer attempts to send announcements to the rest of the nodes in a cluster.
CR081817	There is now a bean-level option to set the `dispatchPolicy` in the `weblogic-ejb-jar.xml` file.
CR082182	Source files in the `ejbc` output directory are no longer unintentionally deleted when running `ejbc`.
CR082350	In 6.1 SP3 a problem caused the EJB Handle Size to increase dramatically. The problem has been resolved
CR082451	Third-party jars are now loaded properly in situations in which the JMS Provider is not WebLogic-specific.
CR082566	Per-bean pessimistic concurrency is now supported with Microsoft SQLServer. Pessimistic locking can be done with the use of use-select-for-update tag in the DD. This doesn't work with MSSQL Server. The equivalent SQL for it would be of the form: SELECT ... FROM ... WITH(UPDLOCK) WHERE ... Adds support to generate 'WITH(UPDLOCK)' in all SELECT queries if database-type tag in DD is SQL Server.
CR082787	In earlier versions making TX_SUPPORTS for 'create()' and 'remove()' methods breaks the basic functionality of locking and unlocking stateful session beans. This problem is resolved by restricting 'create()' and 'remove()' to be run in Unspecified tx context.
CR083050	In WebLogic Server 6.1SP#, When calling context.getCallerPrincipal() from ejbRemove(), an exception similar to the following exception is thrown: `<Jul 31, 2002 4:52:17 PM EDT> <Error> <HTTP> <[WebAppServletContext(4292442,test,/test)] Servlet failed with Exception weblogic.ejb20.interfaces.PrincipalNotFoundException: Error. Method SecurityHelper.getCallerPrincipal returned a NULL Principal. The CallerPrincipal Stack has been corrupted. One cause of corruption might be: A Bean has Created its own Threads. Note that this would be in violation of EJB2.0 Final Spec Chapter: Runtime Management 24.1.2 at weblogic.ejb20.internal.EJBRuntimeUtils.getCallerPrincipal(EJBRun timeUtils.java:526) at ...` This is fixed in this release.

Change Request Number	Description
CR083239	The EJB QL NOT MEMBER clause no longer causes infinite looping.
CR083240	The EJB QL NOT MEMBER clause in conjunction with WHERE no longer creates a bad query.
CR083689	Container-managed persistence automatic key generation now works with SQLServer.
CR084407	In some earlier versions, ExclusiveEntityManager is not returning beans to pool correctly. When beans exited the cache, they were not returned to the free pool as they should be from the ExclusiveEntityManager. This is fixed in this release.
CR084938	Users can now configure the read-timeout-seconds for a read-only bean using an application-scoped cache.
CR084978	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.
CR085320	The "Idle Beans Count" value in the Administration Console is no longer constantly increasing.
CR085903	ejbRemove() is now called correctly during undeployment of a message-driven bean.
CR088223	A problem with the serialization of bean-managed-persistence stateful session beans during passivation has been resolved.
CR088526	In WebLogic Server 6.1 SP3, the EJB container breaks EJB2.0 contract: "Tx not set to rollback when using BMT and exception is thrown". In this release, The container logs the exception (works fine earlier) delete the bean instance (works fine earlier), and mark the transaction for rollback (doesn't work earlier). Also the transaction is disassociated with the thread.
CR090356	Fixed a problem in which the EJB container was parsing security role information incorrectly.
CR093776	ejbc no longer generates syntactically incorrect code when a column is defined as LongString in weblogic-cmp-rdbms-jar.xml.

Examples

Change Request Number	Description
CR055626	The `login.jsp` file for the `webapp security` sample now allows for URL encoding when a user logs in.
CR069680	Fixed a problem in `JdbcTable.jsp` example. It no longer sets the database connection properties for username and password to the literals of the variable names; it now correctly uses the variables content.
CR085521	Fixed a problem in which the `package-summary.html` file of the `r3client` Web services example was incomplete. The file now contains all of the necessary information.

Installer

Change Request Number	Description
CR084521	The Configuration Wizard no longer creates a misconfigured JMSJDBCStore in the PetStore domain.
CR088287	The Configuration Wizard no longer generates an incorrect SSL port when creating Managed Servers in console mode.

Internationalization

Change Request Number	Description
CR074933	Administration Console help no longer displays 404s or displays in English instead of displaying Japanese translation.

jCOM

Change Request Number	Description
CR076584	A `com2java.exe` problem in which generated classes would not compile without manual intervention has been resolved.

JDBC

Change Request Number	Description
CR062827	Prepared statement code no longer provokes a null pointer exception. The problem was that the prepared statement object was being set to null.
CR072605	Prepared statement caching is now off by default, that is, the default statement cache size is now 0.
CR082011	Creating connection pools with the `weblogic.admin` command-line interface no longer results in this authentication error: `java.lang.ClassCastException: weblogic.security.service.PrincipalAuthenticator`
CR083928	WebLogic Server JDBC now supports the `NLS_LANG` parameter `WE8MSWIN1252`.
CR085559	When a user transaction failed to roll back due to the Oracle being down, the JDBC connections used in the transaction were released. As a result, no connections were available in the pool. This problem has been resolved.
CR087803	WebLogic Server can now create a JDBC store configured for a JMSServer using Oracle 9.0.1 version.

Change Request Number	Description
CR087930	Previously, prepared statement cache functionality did not include support for calling `setInt()` for statement columns that were first used with `setLong()`, `setFloat()`, or `setDouble()`. In addition, calling `setInt()` on a column that had originally been bound with `setNull(i, BIGINT)` caused a "Re-Parse Cursor" SQLException. The missing functionality was supplied by the exposure of a new XA Prepared Statement Cache in the JDBC Connection Pool MBean, which is turned on by default with a default value of 5, and by other code changes. The default value for Prepared Statement Cache Size was also increased from 0 to 5, which enables the feature by default. See "Increasing Performance with the Prepared Statement Cache" for more details, including usage restrictions.
CR089185	Transaction isolation can now be set when using jDriver/Oracle XA and Oracle Thin XA. Previously, attempting to set it resulted in this error: `get exception: java.sql.SQLException: Due to vendor limitations, setting transaction isolation for "Weblogic OCI XA" JDBC XA driver is not supported`
CR090378	Running continuous Selects through a JDBC connection pool no longer provokes `OutOfMemory` errors. 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 7.0SP2, 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*.

jDriver

Change Request Number	Description
CR059097	The `statement.cancel` method is now working with WebLogic jDriver for Oracle.
CR068911	`ResultSet.getCharacterStream()` now works for Oracle LONG type.
CR078427	When using a PreparedStatement via weblogic.jdbc.pool.Driver with the Oracle jDriver, its parameters were not cleared properly and jDriver inserted conjunct data (new data of parameter and previous data of parameter) into the database.
CR078936	On the HP platform, attempts to connect to connect an Oracle database using `weblogic.jdbc.connectionPool.oraclePool` no longer result in a `SIGSEGV 11` HP error.
CR082484	The JDriver for Oracle no longer raises an `ArrayIndexOutOfBoundsException` if there are more than 511 parameters in a prepared statement. Limit is now 1024 instead of 512.

JMS

Change Request Number	Description
CR063743	Message-driven beans were not acknowledging object messages; messages are now queueing correctly to the message list.
CR077805	For the null string user property— `setStringProperty(propertyName, null)`—the property name is now reflected in `Message.propertyExists()`.
CR077898	Using a serverSession pool no longer causes `Message.getJMSDestination()` to return null.

Change Request Number	Description

CR078439	In earlier versions the `JMSSession.commit` method threw exceptions similar to the following stack trace, when both client and server were colocated (server-side tests) in the cluster environment. (HP AS and Sol MS)
	
	`EXCEPTION: java.lang.Error: transaction suspended twice Start`
	`server side stack trace: java.lang.Error: transaction suspended`
	`twice at`
	`weblogic.jms.dispatcher.Request.forceSuspendTransaction(Request.`
	`java:874) at`
	`weblogic.jms.dispatcher.Request.sleepTillNotified(Request.java:2`
	`00) at ...`
	
	The problem is fixed in this release.

CR078702	The `JMSDispatcher` thread no longer has a problem with the security context when dispatching messages to a remote server.
CR079944	The handling of JMS Queue browsers has been modified so that associated memory is cleaned up when the browser is closed, even if the entire enumeration of messages has not been traversed.
CR080301	In WebLogic Server 6.1SP2, MessageListener is not receiving messages after TransactedSession Roll back. The message listener appears to stall after processing a certain block of messages and performing a certain amount of rollback() calls. This stalling appears to occur on the second batch fetch from the JMS Queue. In other words, if the Messages Maximum parameter equals 10, the first 10 messages will be processed correctly, however the message listener will stall on the 11th or 12th message. If Message Maximum equals 1, the stall is on the second message and sometimes 3rd message. This is fixed in the current release by properly maintaining the internal data structures with Queue commits.
CR081306	Redelivery of JMS messages for a durable topic subscriber is no longer restricted to batches of 10 and is now configurable.
CR081429	Creating a connection pool using jDriver with Oracle 8.1.7., then configuring a JDBCStore for JMS using this pool, and enabling JDBC no longer throws a "`java.sql.SQLException:` `Connection has already been closed`" error.
CR082826	Durable subscriptions are no longer erroneously unsubscribed when a JMS Messaging Bridge is stopped, undeployed or shut down.
CR083196, CR088255	A JMS Messaging Bridge no longer drops messages if the destination server hosting the messaging queue is shut down.

Change Request Number	Description
CR083290	Recovery of a server in a cluster has been improved to handle cross-server notifications appropriately. Changes were made at the networking level, at the RMI level, and in JMS.
CR083794	Sending messages to a Topic whose `ServerAffinity` property is set to `False` no longer causes a `NullPointerException`.
CR089114	For a certain class of applications, WebLogic JMS can now significantly optimize topic subscriber selectors by indexing them. These applications have messages that contain one or more unique identifiers and thousands of subscribers that filter based on these identifiers.

A typical example is an instant messaging application where each subscriber corresponds to a different user, and each message contains a list of one or more target users. |
CR084175	Messages are no longer lost if the WebLogic Server instance is killed and restarted while the MessagingBridge is processing the messages.
CR084182	Setting a MessageListener or an ExceptionListener to `Null` no longer throws a `NullPointerException`.
CR084374	The error message that displays when a Messaging Bridge fails to establish an initial context with a server no longer includes the Bridge's password as part of the message.
CR085179	Closing a JDBC store no longer causes outstanding JMS I/O requests to be dropped.
CR085766	Calling `recover` followed by `ClearProperties` no longer causes message properties to be dropped.
CR086110	JMS security policies are now checked for message queue browsing; anonymous users can no longer browse a message queue.
CR086125	The selectors code has been optimized, resulting in up to a 30 percent performance improvement when there are high number of subscribers, each with a unique simple selector.
CR086413	Distributed destination messages no longer contain incorrect birth or expire times.
CR086495	JMSServer undeployment ("remove deployment") no longer throws an `InstanceNotFound` exception.
CR086976	JDBC store creation now succeeds on an AS400 configuration running DB2.
CR087131	`Redelivered` flag is no longer set to `True` when it should be set to `False`.

Change Request Number	Description
CR087524	Pending counts for JMS DurableSubscriber are no longer set to negative values.
CR088952	When a JMS client program is disconnected abnormally, associated resources are now completely cleaned up on the server side.
CR089730	`createTopic("BackendName/TopicName")` is now working correctly.
CR090508	Messages no longer erroneously stay on the queue when a message has its `replyTo` set to a distributed destination.

JNDI

Change Request Number	Description
CR083705	WebLogic Server 7.0 has a slight performance regression with JNDI lookups and use of EJBs. This has been fixed by allowing pre-generated JNDI-related stubs. `ejbc` has a new flag—`disableHotCodeGen`— which generates stubs and skeletons during compilation time.
CR085463	In WebLogic Server 6.1 SP3 JNDI lookup failover is taking a long time when the network plug is pulled. This has been fixed in this release.

JTA

Change Request Number	Description
CR064403	Fixed a problem in which after a transaction timeout using XA driver + SSB, the XAConnection became unusable until a server restart.
CR073072	Setting the JTA transaction timeout value to 20,000,000 or larger no longer provokes an exception (`javax.transaction.SystemException: start() failed`).

Change Request Number	Description
CR073649	Calling a connection from the XA connection pool after the transaction has timed out no longer throws an XA-PROTO error.
CR080755	Fixed a deadlock issue with Oracle XA thin driver.
CR090618	The `"direct-write"` file I/O option for JTA and JMS is now available on HP.

JTS

Change Request Number	Description
CR064301	DBAs no longer have to resolve distributed in-doubt transactions manually; WebLogic Server now resolves them automatically.

Miscellaneous

Change Request Number	Description
CR042655	When using `access.log` in Extended Log File Format all times are now specified in GMT instead of in local time.
CR072188	Network connections for performing server life cycle operations between an Administration Server and a node manager are now closed after task completion.
CR073023	The node manager no longer throws an `OutputHandler` error when trying to start a remote server.
CR068646	MBeans are no longer accessible with HTTP despite `HttpdEnabled` being set to false and `TunnelingEnabled` being set to false.
CR071796	A server no longer fails to launch from a non-default directory

Change Request Number	Description
CR079110, CR083257	When a WTCServer run-time MBean was deleted using the weblogic.Admin DELETE command, the command was not UNDEPLOYING the WTC before deleting it. This problem has been resolved.
CR079455	A deployment failure caused a ConfigurationError resulting in the server process dying. This problem has been resolved.
CR080016	Fixed a problem in which starting in an Administration Server provoked a `java.lang.reflect.UndeclaredThrowableException` if the application targeted a virtual host to the Managed Server but not the Administration Server.
CR080324	`WebLogicMBean.setParent()` is now exposed as a public method in Javadoc.
CR080740	A graceful shutdown feature is added for NT Services. There is now a `-stopclass` option for `beasvc`. When this option is set, the `stop()` method on this class is invoked to perform the shutdown.
CR082081	The SERVERLOG command line utility no longer fails when the secure T3 protocol is in use. Previously, this command: `Dweblogic.security.SSL.trustedCAKeyStore=%WL_HOME%\server\lib\cacerts -Dweblogic.security.SSL.ignoreHostnameVerification=true weblogic.Admin -url t3s://localhost:7502 -username system -password password SERVERLOG "2002/07/17 13.00" "2002/07/17 14.00"` Resulted in this message: `Unable to get log file:unknown protocol: https`
CR082910	When registering custom MBeans are registered within an EAR, an attempt to use JMX/MBeans to remove a JDBC connection pool no longer fails with `java.lang.ClassNotFoundException`.
CR083220	Attempts to instantiate a new COMMO MBean no longer throw a `javax.management.MBeanException` on the client side.
CR083400	Managed Server startup no longer fails with a stack overflow exception when the secureT3 protocol is in use.
CR084607	JMS topics and queues are now being created properly at runtime, in a cluster. Previously, it appeared that the topic had been created, but clicking on the Destination Link under JMS Server resulted in these exceptions: `java.lang.reflect.UndeclaredThrowableException` `javax.management.InstanceNotFoundException`

Change Request Number	Description
CR085315	On AIX 4.3, booting a Managed Server, with the Administration Server running, no longer causes the WebLogic Server process to exit with a `NullPointerException`.
CR085914	SNMP requests no longer time out after a Managed Server is dropped.

CR086913	In a Web Logic Platform domain, WebLogic Server no longer fails to recognize the following three properties during startup:
	`weblogic.jws.ProductionMode`
	`weblogic.servlet.ClasspathServlet.disableStrictCheck`
	`weblogic.SystemDataStoreConfigDirectory`

CR088449	WebLogic Server no longer fails to recognize the following properties during startup:
	`weblogic.classloader.preprocessor`
	`weblogic.oci.selectBlobChunkSize`
	`weblogic.PosixSocketReaders`

CR090721	The file descriptor count for the WebLogic `java` process no longer grows out of control to the point where the server process dies.
CR092471	Domain log rotation by time no longer fails when the server is rebooted.

Node Manager

Change Request Number	Description
CR071591	The Node Manager no longer tries to detect Managed Server failures by trying to initiate an HTTP connection. If Node Manager receives a connection failure, the connection exception can now be observed in the Node Manager output.
CR072188	After completing the task, an Administration Server now correctly closes the connection it established with the Node Manager. Previously, it failed to, resulting in the server running out of file descriptors.

Change Request Number	Description
CR079205	The NodeManager now parses `weblogic.policy` correctly; it no longer adds an equals sign when copying the property to the command line of the child process.
CR081183	A race condition no longer occurs when the Node Manager kills a cluster. This problem had been resulting in `java.net.ConnectException: Connection refused: connect` errors.

Plug-ins

Change Request Number	Description
CR079186	All plug-ins now parse folded headers properly as per the HTTP 1.1 specification. Previously, folded headers provoked "malformed header" error messages.
CR087997	The Apache plug-in is now correctly parsing the primary `jvmid` from the session cookie.
CR089033, CR088914 CR088915	Fixed plug-in vulnerability to cert chain attacks, a security risk.
CR079973	Failover is now correctly prevented when `Idempotent` is set to `Off`, for NSAPI. Core dumps no longer occur as a result of setting `ConnectTimeoutSecs` to `0`.
CR080219	IPlanet no longer core dumps when all WebLogic Server instances are shut down.
CR080382	"error page is unavailable" messages no longer appear when page is actually available.
CR082093	The proxy plug-in made spurious extra calls to the application server under heavy load. Now, if a call comes into iPlanet, only one call goes to the application server.
CR082096, CR083386	Failover is now correctly prevented when `Idempotent` is set to `Off`, for ISAPI.

Change Request Number	Description
CR082113	Previously, when the ISAPI plug-in received an error while reading data from the client, it tried to send the incomplete data (request) to the backend WebLogic Server. This situation occurred if, for example, the end user pressed the browser stop button while the plug-in was reading the HTTP POST request
	Because the data being sent did not meet the expected number of bytes, the plug-in received an error in the sendRequest phase.
	This no longer occurs.
CR082206	IPlanet now has the `WLExcludePathOrMimeType` flag, allowing proxying by path but excluding certain images from that proxy.
CR082939	If the URLRewriting property is used for session tracking instead of cookies, stickiness is now maintained. Previously, session would not be tracked properly even when URLRewriting was set.
CR083643	There is no longer a performance problem in NSAPI in which `netbuf_getc()` was very slow and was not able to distinguish between `'\0'` and `0` in the postdata.
CR085192	The KeepAlive feature is now enabled by default for ISAPI and NSAPI plug-ins.
CR085922	In NSAPI `netbuf_getbytes()` no longer fails with a Java client with iWS4.x.
CR086518	Fixed a problem in IPlanet in which, if performing a redirect using a URL that contained both a query string and an anchor, the anchor was incorrectly considered part of the value portion of the last query string member.
CR089746	Session caching and KeepAlive functionality are now enabled for secure sessions.

RMI/RMI-IIOP

Change Request Number	Description
CR079918	Attempts to store RMI objects in a database no longer fail due to a problem with serializing dynamic proxy objects.
CR081265	Fixed a problem with replica lists that occurred when calling into a cluster and was manifesting as an `AssertionError`.

Change Request Number	Description
CR081510	The runtime descriptor for CSIv2 support in the IOR for an RMI object now defaults to `Supported`. Previously, it defaulted to `None`.
CR081764	Fixed an IIOP initialization deadlock that was causing servers to hang on restart under heavy load conditions.
CR081923	Fixed a memory leak in stub generation that was resulting in `IncompatibleClassChange` errors.
CR083018	Fixed a problem in which WebLogic Server could not unmarshal `Any`'s using indirect typecodes. This was resulting in interoperability problems with IBM.
CR083936	Fixed a series of problems that were preventing using WebLogic Server to coordinate a transaction with Orbix 2000 2.0.
CR085703	Requests no longer close prematurely when an Exception response occurs.
CR086300	The domain name field—in which WebLogic Server places the server name—is now set correctly in the replica list in a service context.
CR086350	Fixed a problem in which generated IDL was corrupt when the exception class name terminated in a capital I ('I').
CR088777	Circular references are now handled correctly in IDL generation. Previously, if classes contained circular references, attempts to compile them with the -iiop -idl options failed.
CR088778	Using indirection within TopLink no longer throws Assertion failures.
CR090010	Fixed a problem in which weblogic.ejbc was generating an incorrect IDL module declaration for multi-dimensional arrays.

Security

Change Request Number	Description
CR066491	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-17.jsp.
CR072154	Problems corrected in `sslclient` example: • `Package-Summary.html` no longer contains blank classes. • `WebLogic Server_home` replaced with `Samples_home.` • `MyListener` class is now in the kit.
CR072432	Fixed a problem in which certificate data was not appearing in the Administration Console.
CR072954	SSL is now working properly in applets.
CR073187	The TLS_RSA_WITH_DES_CBC_SHA cipher suite is now exportable.
082040	The location of the log files for the WebLogic Auditing provider has changed. Auditing is configured for the security realm, but each server instance writes to its own log file located in the server directory. The new location is `WL_HOME\mydomain\myserver\ DefaultAuditRecorder.log.`
CR084170	In previous releases, 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. For more information, see SSL Certificate Validation.
CR085072	Defining security policies in the weblogic.xml or weblogic-ejb-jar.xml files required creating a \lib directory in the domain and then granting permissions to the \lib directory in the weblogic.policy file. This known problem is fixed in this release. For information about defining security policies, see Securing WebLogic Resources.
CR085203	WebLogic Integration no longer experiences problems resuming the SSL session when interacting with WebLogic Server.

Change Request Number	Description
CR085805	The code example for the custom Authorization provider did not work because the `web.xml` file in the code example was incorrect. The file used a `url-pattern` of * to protect its pages when it should have used */.
	Updated code examples are available from the following URL:
	http://dev2dev.bea.com/code/
CR086589	The `javax.security.auth.login.LoginContext.logout()` method did not delete the principal and credential information from the subject of a LoginContext. Therefore, the user was not logged out.
	This known problem is fixed in this release.
CR086709	Protected JNDI resources can no longer be accessed by unauthorized users.
CR088081	WebLogic resources with hierarchies (for example, JMS, JDBC, Server) now handle actions correctly from the `getParentResource()` method.
CR088784	Calling a stateful session bean from a JSP caused the following security violation:
	`java.rmi.AccessException:Security Violation: user username has insufficient permission to access EJB`
	This error occurred because an established security context was not closed properly. Use `ServletAuthentication.runAs(Subject subject, HTTPServletRequest request)` at the end of the JSP so that the user will be added to the HTTPServletRequest if the initial security context is not closed.
CR089059	WebLogic Server rejects digital certificates that have the KeyUsage constraint set as a critical, but do not have the KeyEncipherment constraint set.
	Older versions of WebLogic Server which use SSL V3.0 did not reject certificates under this circumstance. However, this release of WebLogic Server which uses TLS V1.0 will reject certificates under this circumstance.
	Workaround: Apply patch `CR089059_70sp1.jar`.
CR090360	The ServerIron monitoring product sends a request to the SSL port during a server health check. During this health check, the SSLIOConect and related SSL objects were not released, causing an out-of-memory error.
	This known problem is fixed in this release.

Change Request Number	Description
CR090401	The users group no longer includes unauthenticated users. This known problem is fixed in this release.
CR091796	Dynamic groups now works properly with the IPlanet Authentication provider.
CR091922	The servlet container in WebLogic Server did not pass login exceptions to the error page as specified by the J2EE specification. This known problem is fixed in this release.
CR092167, CR085441	Please review the security advisory information at http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-29.jsp.
CR077050	Fixed a problem in which, when adding users to replicated LDAP databases in cluster, it took an unacceptably long time for users to be replicated on Managed Servers.

Servlets and JSPs

Change Request Number	Description
CR065967	Fixed a problem in which tag libraries failed when the `type` option was specified and the type was an array; the type was erroneously treated as a string.
CR068577	Fixed a problem in which `method_releaseTags()` was not called as part of `jsp:forward`, in generated code. Tags are now released when a forward occurs within the body of a JSP.
CR070417	Fixed a problem in which, when a servlet sent messages to an applet in HTTP/1.1, the applet would hang until all the messages were sent. That is, the applet did not receive the messages in parallel as the servlet sent them.
CR075419	URLs containing a "%" now load correctly.
CR044926, CR075799	Java code can now be generated from JSP files that include a '}', on 2Byte platforms.
CR072474	When using `jsp:usebean`, `jspc` no longer generates class files with a deprecated method.

Change Request Number	Description
CR075668	`jspc` now works correctly when JSPs are located in directories other than the root directory and expressed in relative pathnames.
CR078551	When `jsp:include` refers to a non-existent JSP, a `FileNotFoundException` is now thrown.
CR081353	`javax.servlet.jsp.PageContext.out` attribute now cleaned up correctly for body content tags; output is now sent correctly to the browser.
CR082310	JSPs with a `page contentType` directive can now be deleted and overwritten in an exploded web archive (WAR).
CR083227	If a JSP file contains a `jsp:setProperty`, and the bean this tag refers to contains a multidimensional array as attribute String [][], `jspc` now generates compilable code.
CR083597	Content type is now set correctly when the `page contentType` directive is used.
CR83804	WebLogic Server now allows the user to reset the buffer size as long as the response is not committed to the client. In 7.0 and 7.0 SP1, the server would throw an `illegalStateException setBufferSize` was called after some `ServletOutputStream.write()`'s, even if the buffer had not been committed. This caused problems with Servlets/Filters which that depended on calling `setBufferSize` before flushing the buffers.
CR083848	Japanese characters in a JSP are no longer corrupted when including content from another JSP.
CR084593	A JSP now correctly detects that `contentType` is set to an illegal character set.
CR085702	Line numbers for JSP compile errors now correctly match up to the error message.
CR086908	`jspc` parsing error messages made more specific to help users quickly locate malformed descriptors.
CR087090	`jspc` execution time is now more efficient, due to how `jspc` calls multiple JSPs that use identical tag libraries.

Change Request Number	Description
CR088019	Fixed a problem in which deployed resources of multiple virtual hosts conflicted due to sharing a common temporary directory. This was resolved by including the server name in the temporary directory path.
	A side-effect is that compile-on-startup JSP classes are not deployed to Managed Servers from an Administration Server. However, this was never intentionally supported, and the correct procedure is to precompile these classes before server startup using `weblogic.jspc` and deploying to the `WEB-INF/classes` directory.
CR088301	Fixed a problem in which an included resource ("Resource A") was not sent to the servlet output stream when dispatched from another resource ("Resource B") that had been forward dispatched inside a JSP `BodyTag`.
CR088525	`jsp:include` now carries `jsp:param` properly if the value is "/////"
CR092089	Fixed a problem in which, when a Web application had a JSP file with a multibyte name, WebLogic Server did not compile it and returned the directory contents.
CR093291	Fixed a classpath problem such that `.jar` files are found successfully when compiling JSPs.
CR061493	When an URL is specified as a JSP name/value parameter to an included JSP, the URL is no longer truncated when a different encoding scheme is specified.
CR070090	WebLogic Server now specifies the `cache-control` appropriately to prevent caching proxies from caching a cookie.
CR074840	Fixed a problem in which, when a proxy with `ppath` and `pathtrim` was used between a client and WebLogic Server and form based-authentication was used, the client would not get redirected to the proceed resource.
CR077780	WebLogic Server is now cleaning up and requeuing properly after dispatching a servlet request, thereby preventing memory leaks.
CR078698	The `access.log` file now correctly rotates according to size.

CR079014	The lifetime of HTTP persistent connections is now configurable on the client side, via the new `setTimeout()` method, which is demonstrated in this code snippet:
	`URL u = new URL(url);`
	`HttpURLConnection hu = new HttpURLConnection(u);`
	`hu.setTimeout(5*1000);`
	`BufferedReader in =`
	`new BufferedReader(new InputStreamReader(hu.getInputStream()));`

Change Request Number	Description
CR079767	Temporary files stored in the `applications/.wlnotdelete` file are now deleted when they are no longer needed—when an application is redeployed or removed.
CR80090	If defined, `FrontendHTTPPort`, `FrontEndHTTPSPort`, and `FrontEndHost` now take precedence over a `HOST` header when constructing a redirected URL.
CR080384	Fixed a problem in which `HttpClusterServlet` stopped forwarding requests to a server if a malformed status line was detected in one request.
CR080613	Fixed a problem in which commas in a quoted value in a cookie would not parse correctly.
CR080751	Fixed a problem in which, if an application threw a customized exceptions that extended `ServletException`, the wrong message would appear.
CR080855	An HTTP request containing `absolutURI` now executes with the correct context path, request URI and request URL.
CR080903	Fixed a problem in which Web applications were being extracted twice during deployment.
CR081071	Malformed processes now exit gracefully, rather than throwing a `NullPointerException`.
CR081253	The server now correctly parses HTTP requests with `TransferEncoding` set to `Chunked`.
CR081418	The `PrintNulls` option is no longer missing from `weblogic700-web-jar.dtd`.
CR081484	Session attributes are no longer lost when using `HttpSessionListener` with `PersistentStoreType` set to `replicated`.
CR081521	CGI output now correctly displays when calling a CGI script from a JSP in an `.ear` or `.war` file.
CR082156	isRequestedSessionIdValid() always returned the value, true, 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.
CR082580	When accessing a URL, HTTP `POST` parameters are now correctly preserved when a form-based authentication is invoked and authentication is successful.
CR083191	Fixed a problem in which the Console displayed invalid values for servlet execution times.
CR083487	CGIServlet now work when `<url-pattern>` is an exact match.
CR083517	`HttpClusterServlet` now correctly proxies a request to SSL when `SecuryProxy` is set.

Change Request Number	Description
CR083624	An HTTP 1.0 request now correctly returns an HTTP 1.1 response in the header.
CR083912	Fixed a problem in which JDBC session persistence was failing when using Oracle Thin Driver 8.1.7.3.
CR084002	204 HTTP responses no longer include a message body, as mandated by specification.
CR084076, CR086280	Fixed a problem in which while accessing a servlet over a `Keepalive` connection created a malformed request.
CR084165	HTTP session tracking is now maintained when `PersistentStoreType` is set to `jdbc`.
CR084167	Delegating a JSP's output to another JSP by including or forwarding no longer provokes an `IllegalStateException` when using a response wrapper.
CR084649	WebLogic's CGI server now provides the `SERVER_URL`, `HTTP_COOKIE`, and `QUERY_STRING` environment variables to scripts running within it.
CR084785	`GenericProxyServlet` now handles wrapped responses.
CR084958	The `HttpURLConnection` handler now implements getErrorStream(), thus making it possible to read custom response from the 'error' page.
CR085735	Fixed a problem in which `JSESSSIONID` was encoded in the URL even when `URLRewritingEnabled` was set to `false`, resulting in a 404 error.
CR085843	`ejb-local-ref` now works for Web applications.
CR086026	`CacheSessionCookie` now defaults to true, preventing display problems on the first displaying attempt.
CR086052	Outbound (from WebLogic Server) HTTPS connections no longer require a trailing slash '/' to work.
CR086099	A `NullPointerException` bug resulting from change of behavior of `java.net.Socket.isConnected()` between Java 1.3.1 and Java 1.4.0 has been fixed.
CR086416	A servlet engine no longer throws a `SocketWriteError` when trying to load a console page.
CR086481	A Web application's attempt to upload a file no longer provokes an `ArrayIndexOutOfBounds` exception; the code has been rewritten to handle buffer overflows and other scenarios in which a buffer is only partially read.

Change Request Number	Description
CR086677, CR087573, CR088528	Fixed a problem with JDBC persistence in which subsequent requests were allowed before session persistence had completed, resulting in data loss.
CR087647	`HttpURLConnection` now times out correctly.
CR087984	Fixed a problem in which invoking `ServletContextListener` before `HttpSessionListener` caused unexpected results during server shutdown.
CR088166	Server now more robust at detecting missing XML close tags, thereby avoiding exceptions.
CR088478, CR086490	The `MaxSkips` feature has been deprecated in WebLogic Server plug-ins for clustered servers and replaced with `MaxSkipTime`. `MaxSkipTime` sets the amount of time after which the plug-in will retry the server marked as "bad," where "bad" denotes a server that failed previously.
CR089229	Fixed a problem that was causing 10% performance regression in `HttpClusterServlet` calls.
CR089923	Fixed a problem in which `jspc` was unable to create a file in the current directory.
CR090665	A servlet filter no longer needs to call `flushBuffer()` explicitly on the response wrappers after having called `chain.doFilter()`, in order to send a response to a client.
CR091699	`response.sendRedirect()` now correctly redirects to a relative URL instead of an absolute URL.
CR091878	Undeploying a Web application no longer provokes an `IndexOutOfBoundsException`.
CR092778	Fixed a problem in which `JISAutoDetect` encoding for `HttpRequest` was provoking exceptions.
CR095166	Cookie parsing is now fixed in `iisproxy`: when a client sends multiple cookies the proxy now correctly sticks to the primary server.
CR080778	Turning on session monitoring no longer results in increased traffic and Administration Server hangs each time a new session is created.

Tools

Change Request Number	Description
CR073688	WebLogic Builder now supports entering a value for both Servlet name and URL pattern in Servlet/Filter Mappings->Filter Mappings->Add.
CR073715	If a `.jar` file contains more than one container-managed persistence (CMP) descriptor file, WebLogic Builder now displays all of the descriptor files; previously, Builder displayed only the first one it found.
CR073877	When WebLogic Builder creates a container-managed persistence entity bean, it now defaults the `<trans-attribute>` value to `Required`.
CR074052	WebLogic Builder now correctly displays `enable-call-by-reference` default value as `True`.
CR074059	WebLogic Builder now provides a feature to export deployment descriptors to a user-specified destination.
CR074208	Adding a new finder/select works properly now when setting resultTypeMapping in WebLogic Builder.
CR074246	Fixed a problem in which WebLogic Builder failed to preserve timestamps when originally opening up an archive. In some cases this problem caused the EJB container to erroneously pre-compile all JSPs.
CR074247	Fixed a problem in WebLogic Builder in which adding a finder with the same method name as that contained in an existing finder updated the tree but not the table.
CR074602	WebLogic Builder now reloads classes before validating them.
CR074603	WebLogic Builder now validates classes before deploying them.
CR074634, CR074955, CR074972	Using WebLogic Builder's Relations wizard to edit the relations of a CMP with multiple relations already configured no longer results in an exception.
CR074639	When you edit a Finder method name using WebLogic Builder's editing panel, the change now correctly propagates to XML. When you edit using the Finder pop-up dialog, the change now correctly propagates to XML and to the editing panel.

Change Request Number	Description
CR074876	WebLogic Builder now saves and properly propagates server connectivity information in the Options dialog. Previously the values in the Options dialog had no correlation to the value that would appear in the "server connect" dialog. With this fix, that information is now kept in sync with the values entered in the Options screen.
CR075005	Specifying an invalid drive when opening a component (type bogusDrive in the text box, press enter and select open) in WebLogic Builder no longer causes a NullPointerException.
CR075006	WebLogic Builder now properly constructs one-to-many relations.
CR080967	WebLogic Builder now correctly parses and edits the `weblogic-cmp-rdbms-jar.xml` file.

CR081296	Fixed a `DDInit` problem that was resulting in a `NullPointerException`. Issuing this command:
	
	`java weblogic.marathon.ddinit.EJBInit project.jar`
	
	where `project.jar` is a well-formed `ejb jar` no longer provokes this error:
	
	`Exception in thread "main" java.lang.NullPointerException`

CR081483	WebLogic Builder now defaults the concurrency strategy to `Read` if there is no value set for `verify-column`, thereby preventing an error.
CR082562	WebLogic Builder now supports `use-select-for-update`.
CR083151	Command-line `DDInit` now works for EAR files.
CR084570	DDInit now creates `weblogic-application.xml` as well as `application.xml` for EAR files.
CR085493	EJBGen now has a `global-role` tag.
CR086073	If a server dialog is cancelled, WebLogic Builder no longer erroneously displays an error message.
CR086546	DDInit now correctly determines that two relations that should be equal are in fact equal.
CR089867	WebLogic Builder now supports `LongString` and `SybaseBinary` types for `dbms-column-type`.

WebLogic Tuxedo Connector

Change Request Number	Description
CR077476	Fixed a problem in which WTC transactions were rolling back unexpectedly, due to incorrect handling of `TPEV_SENDONLY`.
CR078774	The processing of messages by the WTC/tBridge with a connection that pulls messages from Tuxedo /Q no longer stops pulling messages after some time.
CR082312	A tpcall with the `TPNOBLOCK` flag set no longer fails.
CR084435	Fixed a problems that occurred in failover and failback situations when the connection policy was set to `ON_DEMAND` or `ON_STARTUP`.
CR084903	The WebLogic Tuxedo Connector `simpappcns` example now supports a multi-platform environment.

Web Services

Change Request Number	Description
CR070387	Non-standard mappings of datatypes are no longer ignored.
CR073899	WebLogic Web services no longer adds extra non-CDATA characters when generating a SOAP fault.
CR076625	Users can now specify a portTypeName via a deployment descriptor element.
CR076706	Fixed a problem with an xml node's parsing of white-space strings.
CR077855	Fixed incorrect `content-type` setting that was resulting in `NullPointerExceptions` when Java clients accessed Web services.
CR078115	Selecting "Define policies and roles for individual service" in the Administration Console no longer throws a `NullPointerException`.
CR081938	Added a Web services example that handles SOAP with attachments without the use of handlers.
CR082626	Introduced more efficiency into serialization resulting in a performance improvement.

Change Request Number	Description
CR082779	`HttpsURLConnection` now correctly picks up identities from `SSLContext`.
CR083278	Corrected a problem that resulted in JAX-RPC client-stub generation not operating properly.
CR083410	Custom `SoapfaultException` is now correctly embedding the detail field.
CR083813	JavaBean array types are now handled correctly.
CR083919	XML data binding no longer fails on inherited types with duplicate elements.
CR084168	WSDL generation now handles recursive types.
CR084220	A WebLogic dynamic client can now invoke a style `style="document"` WebService.
CR084591	Fixed handling of month and day formats in serialization and deserialization.
CR085214	Fixed a problem in which passing dates as parameters resulted in a `java.Lang.IllegalArgumentException`.
CR085246	XSD enumerations are now recognized in WSDL generation.
CR085888	WSDL with schema imports that contain annotation documentation no longer fail.
CR086231	Fixed a problem in which duplicate exception messages were being issued in generated WSDL.
CR087883	The content length is now correctly set in a webservice response for large values with an element of type `xsd:base64Binary`.

XML

Change Request Number	Description
CR080961	Fixed a problem in Xerces1.3.1: an `org.xml.sax.helpers.AttributesImpl.removeAttribute()` call was failing with a SAX error.

Change Request Number	Description
CR081372	`DocumentBuilder.parse()` is now multi-thread safe when the input encoding is Unicode.

CR088349	Requests from a Java webservices client to a WebLogic Workshop-generated webservice that used character encodings that were not UTF-8 or a subset of UTF-8 failed with this client-side exception:
	
	 `javax.xml.rpc.soap.SOAPFaultException: Invalid request`
	
	 and this server-side exception:
	
	 `java.io.CharConversionException: Malformed UTF-8 char -- is an XML encoding declaration missing?`
	
	 Research revealed that the client specified character encoding information only in the HTTP header, not in the XML declaration. The Workshop webservice behaves according to the XML declarations encoding, which is UTF-8 if nothing is specified. If the requests contains characters that are not valid UTF-8 characters, it fails.
	
	 The client was changed to include encoding information in both the HTTP header and the XML declaration. This resolved the problem.

WebLogic Server 7.0 Service Pack 1 Solutions

The following sections describe problems that were resolved for the release of WebLogic Server 7.0 Service Pack 1.

Console

Change Request Number	Description
CR068366	Changing the console context path now has the appropriate effect on deploying the console application.
CR069887	Fixed a problem in which a `load.mlet` file was created in a `myserver` directory. This occurred when creating a new domain using weblogic.Server in a folder that was different from myserver. Now the `load.mlet` file is deleted and the `myserver` directory is no longer created.
CR076739	Fixed a problem where the object class, object hash code, and object to string fields are blank when viewing the JNDI tree for a local stateless session EJB deployment.
CR077126	Fixed a problem that user is unable to configure the `weblogic.xml param ConsoleMain Attribute` for servlet monitoring in the Console.
CR081366	Fixed a problem that prevents users from deleting a Machine object that has one or more configured servers.
CR082942	Fixed a problem that caused a null pointer exception to be thrown when the user's password was changed in the Console and then `continue` was clicked.
CR083460	Updated WebLogic Server to reflect that Netscape is supported on HP_UX.
CR083578	Increased the size of the Max Message Size text field on the Server Protocols tab in the Console to display all digits entered.
CR084114	Fixed a problem on the WebApp Monitoring tab in the Console that caused an exception to be thrown when monitoring the performance of Web applications.

Core Server

Change Request Number	Description
CR055987	Implemented timeout option for HttpURLConnection.
CR066230	Enabled users to set the name attribute, set in the `config.xml` file to match the domain name. If you manually edited the `config.xml` file and added a `Log` element to the domain with a `FileName` attribute that was not the same as the domain name, the server created and used a domain log with the same name as the domain.
CR074835	Added changes to WebLogic Server to accommodate changes in java.net.HttpURLConnection.
CR074888	Fixed a problem that caused an Error 503 during an HttpSession fail over when the primary server is being shut down in a two-server cluster.
CR075321	Fixed a problem that Operator role could not issue `START` command using the `weblogic.Admin` command. A `NoAccessRuntimeException` occurred.
CR077831	Allow local loading of javax classes. This fixed problem that occurred when the AppletArchiver utility was used to generate a client JAR file. If the applet in question used the `java.swing.JApplet` in its code, then the Applet Viewer produce an `output.jar. java.io.FileNotFound` error
CR077919	Fixed a problem in the method `replaceObject()` that caused WebLogic Server to change the original object and the reference to `LeasedRemoteReference` if the object is of type `StubInfo` and the reference is `LocalServerRef`.
CR079738	Fixed a problem that occurred when the references to the large memory structure were static. To fix the problem, null was assigned to the classloader references WebAppComponent and WebAppServletContext.
CR080324	Fixed a problem that caused an error when calling a newly created destination mbean from an array of destination mbeans on a JMS server because `setParent` is not exposed to public API.
CR080779	Fixed a problem that caused the Administration Server to hang every time a session is created after turning on session monitoring.

Change Request Number	Description
CR081193	Fixed a problem that caused logging to Stdout to stop after WebLogic Server has been running for a few hours. No exceptions were thrown and the server continued to function. In addition, ctrl-break stopped working in the windows environment; on a Unix platform, kill -3 stopped producing a thread dump.
CR083485	Fixed a problem that occurred when using t3 as the default protocol. This problem caused the t3 client to hang in JavaSocketMuxer. The fix ensures that the t3 client uses the correct incoming connection instead of bootstrapping to the default connection.

Deploy

Change Request Number	Description
CR070498	Users can no longer delete application Mbeans if the user has not deleted the actual application.
CR080401	Fixed a problem that prevented multiple context roots from working properly.
CR080914	Enabled use of resource-descriptor with application scoped connection pools. This allows WebLogic Server to check if a JNDI name is defined in the `resource-ref-name` element, perform a lookup at the local level and if the DataSource exists, bind it into the comp/env.
CR081311	Fixed a problem that caused a check on the DTD for the `web-app.xml` file to produced a SAXParseException of informational severity level instead of the appropriate warning severity level.
CR082195	Enabled `weblogic.Deployer` to allow an Administrator to successfully deploy all available member of a cluster, even if the cluster has some members that are not available.
CR082263	Fixed a problem that caused `weblogic.Deploy` not to update a web app WAR file deployed from a Windows 2000 system to a server running on a Solaris system.
CR083179	Improved the WebLogic Server exception message that displays when the `resource reference` deployment descriptors in the `web.xml` and `weblogic.xml` of a Web application have out of sync `resource-ref` and `resource-description` elements. Previously, an exception message displayed that did not contain the actual `resource reference` name that caused the problem.

EJB

Change Request Number	Description
CR050001	Provided EJB-QL support for multi-byte characters that fixed problem with the EJB-QL where clause '=' and 'IN' failed to handle fixed Kanji stings.
CR057074	Fixed a problem where the value for beans-in-use-number was greater than the beans-in-cache-number. This information displays in the monitoring section of the Console.
CR060965	The parser now allows spaces around the '=' symbol in an EJB QL query.
CR073297	Fixed a problem that caused message-drivewn beans that attempted to reconnect to their destinations on failure. This causes the creation of a log message for every reconnection failure. Reconnection was tried every 10 seconds and this was causing the log file to be filled.
CR074781	Fixed a problem that caused automatic table generation not to work with the PointBase database.
CR075219	Roles and policies are now enforced on EJBs.
CR075273	Fixed a problem where the DDInit for a WAR file did not print or create a `weblogic.xml` file. It did print the `web.xml` file, but did not create it in the WEB-INF directory.
CR077104	The EJB 2.0 container-managed relationship (CMR) one-to-many relationships between read-only entity beans with Required transaction attribute no longer generates `IllegalStateException` when trying to access the Collection.
CR077986	Wildcard "*" now works in the EJB 2.0 ejb.jar.xml file for <unchecked/> of method-permission.
CR078297	Container-managed persistence (CMP) caching now works as expected when setting the concurrency strategy to optimistic. Previously, instead of updating the cache, it seemed that ejbLoad was called on the second transaction and this means that the database was updated with the external change.
CR078350	Fixed a problem where the query generation caused the CTS failure because the generated query was incorrect.

Change Request Number	Description
CR079086	A message-driven bean that uses an LDAP server for the JNDI information no longer fails to deploy with a `javax.management.MalformedObjectNameException`.
CR079471	SelectDistinct now works with dynamic queries. Dynamic queries that have SelectDistinct in their EJB QL select clauses were returning duplicates. With Select Distinct, the EJB Container weeds out duplicate results.
CR080115	A null value is no longer returned when the user prints the CMP Clob value.
CR080783	Ensured that the EJB marks and handles the Global Security Roles. When a `role-name` element is to be assigned globally named principals, the `global-role` element should be defined in a `security-role-assignment` stanza in the `weblogic700-ejb-jar.xml` file.
CR080920	Added a runtime flag to allow for fully delegated Security checks. A security check was not performed. Policies set in the Console for JDBC Resources were not taking effect, an inappropriate JDBC Resource was being set by the code, and there was no way to set an Administration Resource for creating a new pool.
CR081398	An EJB 2.0 CMP bean with a one-to-many relationship between beans with exclusive strategy and cache between transactions no longer generates an illegalException.
CR081807	Fixed a problem that occurred because `ejb.Deployer` hard wired the `replicatebindings` to `true` on the context object before binding `home` to the JNDI tree. Therefore, when WebLogic Server tried to send announcements to other nodes in the cluster a `NameAlreadyBoundExcedption` occurred.
CR082406	When using Weblogic Server with a Sybase database and a Sybase XA Driver, the driver no longer throws an implicit conversion error when the CMP field of an EJB is binary.

Change Request Number	Description
CR082451	Fixed a problem where third party JAR files for message-driven beans are not properly loaded and a ClassNotFoundException occurred in situations where the JMS provider is not WebLogic specific.

CR082749	A new tag has been added to the `weblogic-rdbms20-persistence-700.dtd` in WebLogic Server 7.0 SP1, called `use-select-for-update` that enforces pessimistic concurrency on a per bean basis. The description of the flag is:
	
	`<!--`
	`Specifying "true" for this flag will cause SELECT ... FOR`
	`UPDATE to be used whenever the bean is loaded from the`
	`database. This is different from the transaction`
	`isolation level of TRANSACTION_READ_COMMITTED_FOR_UPDATE`
	`in that this is set at the bean level as opposed to at`
	`the transaction level.`
	`Valid values are "true", "True", "false" or "False"`
	`Default: False`
	`Used in: weblogic-rdbms-bean` Since: WebLogic Server 7.0 SP1
	`-->`
	`<!ELEMENT use-select-for-update (#PCDATA)>`

Examples

Change Request Number	Description
CR073117	Revised RMI-IIOP example to update the out of date information for ejb cppclients. Changed the DefaultGIOPMinorVersion="1" in the server mbean to DefaultMinorVersion='1' in the iiop mbean.
CR074373	Fixed typo in the JMS sample example where in the Configure WebLogic Server section of the examples instructions, after step 10, -appsdir was missing from the JAR name.
CR075073	Completed a security provider example and posted it to BEA's online Developer Center for WebLogic Server at dev2dev.
CR077817	Fixed a problem where the default URL in the PointBase Console is incorrect. The default URL was missing a colon after the server.
CR078175	Fixed a problem where the instructions for WebLogic Server Petstore reference the incorrect directory for the Petstore EAR files.
CR079455	Fixed a problem that caused a fatal initialization exception during Examples server start-up when the examples-dataSource-demoPool datasource was changes to oraclePool. This problem occurred when running the EJB examples.
CR080073	Upgraded PointBase files to fixed the problem that caused the PointBase server to hang when doing a CLOB update with multi-byte character set data in the application data.
CR080501	Included the Web Services examples in the WebLogic Server product distribution.
CR082913	Fixed a problem that caused errors and exceptions in the wtc.tBridge example. Updated the javadocs to provide instructions on setting the JMS server before running the example.

Internationalization

Change Request Number	Description
CR063067	Internationalized the WebLogic Builder menus, actions, and documentation including the AppFrame and MainAppFrame menus and actions in the Console.

JDBC

Change Request Number	Description
CR080487	The JTD driver no longer leaks pool connections if DBMS rollback fails.
CR082072	Fixed a problem where the AutoCommit behavior for the JTS connection does not work.
CR082964	Provided the following WebLogic Server support for Oracle 9.2: • WebLogic Server jDriver support for Oracle 9.0.1 • WebLogic Server jDriver support for Oracle 9.2 (already certified) • Existing Oracle provided thin driver bundled in WebLogic Server for Oracle 9.0.1 • New Oracle provided thin driver bundled in WebLogic Server for Oracle 9.2 (Pending Oracle approval)

JMS

Change Request Number	Description
CR063743	Message-driven beans of type Object now acknowledge the object messages.
CR075514	Fixed a problem on the WebLogic Messaging Bridge tab in the Console where the IdleTimeMaximum attribute label was shown "In Milliseconds;" it should have been shown in the correct measurement of idle time as "In Seconds".
CR075894	The Delete icon was missing on the Monitor Durable Subscriber table. Therefore, there was no way to delete a durable subscriber from a topic using the Administration Console. The icon has been added.
CR077814	A memory leak no longer occurs with `weblogic.jms.backend.BEMessageReference` when the client runs.
CR078833	Added new TLOG and JMS file store write options. Command line -D parameters for disabling synchronous writes in JMS file stores "SynchronousWritesEnabled?" has been deprecated in this release. The preferred method is to use the new Console or mbean configuration.
CR079855	Provided documentation to tell users that they cannot configure their own General Bridge adaptor. The instructions say that users with non-JMS messaging products need to obtain a third-party OEM vendor or need to contact BEA Professional Services to obtain access to non-JMS source or target destinations.
CR080078	A JMS message's redelivery limit to no longer remains unchanged after a message fails to be delivered to the correct destination.

JNDI

Change Request Number	Description
CR081839	A `NamingException` no longer occurs when obtaining an InitialContext() from a T3Client.

JTA

Change Request Number	Description
CR059175	An XAException no longer occurs when an EJB executing an SQL statement is followed by an RMI call inside the same transaction.

JTS

Change Request Number	Description
CR064301	Fixed the problem that caused a distributed in-doubt transaction to be created when an application was abruptly killed. When the application was restarted, the distributed in-doubt transaction continued to exist and had to be manually resolved by the DBA.

Miscellaneous

Change Request Number	Description
CR067750	Added a check to determine if application scoped pool names and global pool names duplicate each other.
CR068158	Updated the SNMP MIB Reference documentation to provide description of new WebLogic Server SNMP mbeans and attributes.
CR067987	Marked all references to File T3 as deprecated in the WebLogic Server documentation.
CR072268	Provided support for complexType inheritance from simple types.

Change Request Number	Description
CR072949	Checks for an exception message to interpret if a failure is genuine or a false negative from the native layer. The problem is that the Node Manager tries to do a `forceShutdown` command on a Managed Server that can result in a error being reported back to the user and a misleading message being written to the Node Manager log.
CR073988	Fixed the problem that caused the Node Manager not to resume monitoring Managed Servers that were hosting migratable services.
CR074006	Fixed a problem that caused some mbeans attributes to have the same oid.
074653	Fixed a problem in the WebLogic Server Administration Console's Deployment Descriptor Editor that replaced "&" with "&amp" when writing to memory/disk because "&" is a special character which must be escaped as "&amp" in XML files.
CR075521	Fixed search problem in the online help by deleting a duplicate file that caused the wrong file to display during some searches.
CR075538	Provided documentation that covers the specific Server states that the Node Manager defines as follows

Node Manager defines its own, internal Managed Server states for use when restarting a server. You may observe these states when working with Node Manager in a domain:

- FAILED_RESTARTING—Indicates that Node Manager is currently restarting a failed Managed Server.
- ACTIVATE_LATER—Indicates that MaxRestart restart attempts have been made in current RestartInterval, and Node Manager will attempt additional restarts in the next RestartInterval.
- FAILED_NOT_RESTARTABLE—Indicates that the server is Failed, but Node Manager cannot restart it because the server's AutoRestart or AutoKillIfFailed attribute is set to False.
- FAILED_MIGRATABLE—Indicates that the server is Failed, but Node Manager cannot restart it because the server's HostsMigratableServices attribute is set to True. |

Change Request Number	Description
CR078272	Fixed a problem that prevented iPlanet plug-in from working on the Tru54unix platform with 64-bit support. To use 64-bit support, make sure that you install the following two binaries for iPlanet 6x plug-in and replace the existing ones for iPlanet 4x plug-in: • libproxy.so • libproxy128.so Contact BEA support to obtain these binaries.
CR078782	Fixed a problem where the Connector descriptor code did not obey the encoding for the `weblogic-ra.xml` file. Instead, it was reading the data in the default encoding for the JMV. Now, the encoding specification in the XML header is handled appropriately for deployment descriptor files and RAR files.
CR079186	Enabled all plug-ins to parse folded headers properly as required by the HTTP 1.1 specification.
CR079462	Ensured that details in application-scoped pools is handled in the `weblogic-application.xml` file. This fix ensures that if local properties are absent that global properties are used. Otherwise, if driver, url, and connection properties are not present in the jdbc-connection-pool stanza of the `weblogic-application.xml`, the definitions in `JDBCDataSourceFactopry` are not necessarily used.
CR079672	Provided solution to the problem where the `WarClassFinder.getSource` could lead to a JVM crash.
CR079683	Fixed a problem that occurred with Apache plug-in with CTE when using `jsp.include`. Chunked Transfer Encoded HEX numbers are displayed to the browser when using `mod_wl_ssl_so`.
CR079831	Fixed a problem wherein the 7.0 message catalog DTD did not include loggable attributes as the DTD was not updated to accommodate this feature. This was not a problem for the message catalog parser, but was an issue for the generic XML editors.
CR079973	Fixed a problem that caused multiple attempts to connect to server with Idempotent set to OFF and with ConnectTimeoutSecs set to zero by preventing failover if idempotent is OFF and prevent core dump ConnectTimeoutSecs is zero.

Change Request Number	Description
CR080016	Fixed a problem that occurred when deploying an application while starting a WebLogic Administration Server, if the application is targeted to a virtual host, which is targeted to a Managed Server and an Administration Server.
CR080073	Fixed a problem that caused the PointBase server to hang during a CLOB update when multibyte character set data is used in application data.
CR080219	Fixed a problem that caused a core dump with iPlanet when shutting down all instances of WebLogic Server.
CR080249	Fixed a problem that caused the WebLogic Server 7.0 Administration Server, when set as an NT Service, to be unable to talk to the Node Manager.
CR080257	Fixed a problem that caused a NegativeArraySizeException to occur when the first line of an HTTP request was not formatted properly.
CR080382	Fixed a problem that caused unexpected error message to be logged in the iPlanet Web Server's error log when using the NSAPI plug-in with the following setting: ErrorPage=[any]
CR080746	Provided support for 64-bit platform on native webserver plug-in.
CR080931	Fixed Oracle jDriver problem that prevented PreparedStatement from working with min_bind_size after using CallableStatement by modifying db\oci\OciConnection.java.
CR081377	Fixed a problem that occurred when loading classes via the context class loader during deployment.
CR081625	Included PointBase 4.3 in WebLogic Server.
CR082081	Fixed a problem that caused the weblogic.Admin SERVERLOG command line utility to fail when t3 is used.
CR082093	Fixed a problem with WebLogic Server and the NSAPI plug-in that caused several calls to be made to the application server for each call arriving in proxy.
CR082096	Fixed a problem that occurred when Idempotent is set to OFF that caused the IIS proxy plug-in to attempt one retry for all POST requests after the server.

Change Request Number	Description
CR082113	Fixed a problem that occurred when the WebLogic Server-ISAPI plug-in received an error while reading data from the client and then tried to send the incomplete data (request) to the backend WebLogic Server.
CR082939	Fixed a problem with ISAPI plug-ins so that it works with different session ID formats.
CR083151	Fixed a problem with `weblogic.Builder` that caused the command line ddinit for EAR files not to work.
CR083174	Fixed a problem that caused the latest IIS plug-in not to work when adding new values to the session. For example with the IIS, the method `KeepAliveEnabled=true` does not failover properly.
CR083643	Improved performance with the NSAPI plug-in when trying to read post data. Problems occurred with distinguishing between `\0` and `0` in the post data.

RMI/RMI-IIOP

Change Request Number	Description
CR073691	Implemented tracing feature for RMI-IIOP.
CR074790	Fixed the problem that caused AccessControlExceptions when using RMI-IIOP inside Applets.
CR076974	Fixed a problem where `weblogic.rmi.ReplyOnError.execute` fails when there is no request.
CR079387	Fixed a problem where subcontext is broken in RMI-IIOP. Looking up a subcontext in JNDI causes the root context to display instead.
CR079439	Fixed a problem where strings passed via Any methods are encoded incorrectly. This problem occurs when you pass a String to a remote function that has a declared type of Object, Serializable or Externalizable.
CR079454	Fixed a problem where the interop listBindings is completely non-functional.
CR079645	CORBA protocol exceptions such as `OBJECT_NOT_EXIST` are being mapped erroneously to RMI-IIOP `java.rmi.MarshalException`.
CR079648	Fixed a problem that caused no IOR to be defined for `TransactionFactory` which caused users to be unable to get the OTS `TransactionFactory` from a client.
CR079918	Fixed a problem that occurred because the dynamic proxies were not serializable. This happened when RMI objects were used on each client to facilitate a callback request from a EJB to a client.
CR080239	Fixed a problem that caused the IDL Package names to not be managed properly.
CR080259	Fix problem that caused the component type for boxed RMI sequences to generate wrong sequence of valuetypes.
CR080306	Fixed a problem where under some circumstances, valuetype exceptions were missing their inherited base classes.
CR081135	Fixed a problem caused when large service contexts in a request break fragmentation.

Change Request Number	Description
CR081923	Fixed an RMI/IIOP stub code generation problem that caused an `IncompatibleClassChangeError`. The generated stubs were not handling the ensuing `RemarshalException` correctly.
CR083018	Fixed a problem where WebLogic Server cannot unmarshal `Any`'s method types using indirect typecodes.

Security

Change Request Number	Description
CR069969	Created a new algorithm to allow WebLogic Server to load trusted CAs. Previously, if users did not configure trusted CAs, WebLogic Server loaded a default trusted CAs from the JDK cacerts file.
CR070307	Fixed the server failure problem caused when the user starts a server that is configured for two-way SSL.
CR071894	The default keystone now logs an informational message if it has a private key or root CA keystore configured, but the file does not exist or is not present in the specified location.
CR072902	Modified the algorithm for loading trusted certificates.
CR073082	CertGen no longer adds the domain name extension on the common name field in the certificate when generating certificates on Solaris. Once the `HostNameVerification` is enabled on Solaris, it fails due to a mismatch in the hostname.
CR073665	A `ProtocolException` no longer occurs when tunneling HTTPS through the iPlanet Proxy Server.
CR073941	Enabled the default keystore to log error messages.
CR074102	Fixed a problem that caused the Administration Console to no longer uses a default cookie name.If you tried to load the console through another Web application you were not be able to access it with a single sign-on unless you configured your Web Application to have the same cookie as the console: `ADMINCONSOLESESSION`. Consequently, this caused a cookie name conflict between the console and the WebApp's single signon (SSO) mechanism.
CR074612	Fixed a problem that caused SSL not to start when WebLogic Server is started because the `cacert` file specified in the jdk directory is missing and the location of the trusted `cacert` file is specified on the command line.
CR075172	Fixed a problem caused when SSL handshake fails to find the CA certificate in the configured root KeyStore.
CR075901	Fixed a problem of how the Console handled lists when displaying users while the embedded LDAP was loaded with 50000 users.

Change Request Number	Description
CR076409	Fixed the problem that caused an NTSocket Muxer crash. This problem surfaced when using a Perl script that created connections to the server. The script caused the web server to crash.
CR076945	Fixed a problem that occurred with starting WebLogic Server due to possible corruption of LDAP files. This problem often caused a java.lang.OutOfMemory error
CR078249	Fixed a problem that caused nested groups in LDAP V2 Realm to no longer work.
CR078797	Fixed a problem where when the EJB Container checks for Method Permissions on all methods (except those that are not excluded or unchecked), user is denied execution permission on a method on which no Security Restrictions are set.
CR078887	Provided a way to disable anonymous login to the embedded LDAP server so that anonymous login is prohibited.
CR080072	Fixed a problem that caused the 7.0 version of WebLogic Server with a 7.0 SSL configuration that included the private key keystore, alias, and password specified in the `config.xml` file to default to the 6.x SSL configuration.
CR080490	Fixed a problem where the `DefaultKeyStore` provider doesn't support non JKS JDK Keystores.
CR080508	Updated WebResource to enable support for fine-grain servlet security.
CR080567	Fixed a problem that caused error messages from LDAP to display in the Administration Server log. The VDE server now replicates correctly after change log truncation.

Change Request Number	Description
CR081494	The Role Mapping and AccessDecision Security Service Provider Interfaces (SSPIs) can now access the HttpServletRequest and HttpServletResponse objects via a Context Handler. For example: ```\nHttpServletRequest req = (HttpServletRequest)\ncontext.getValue("HttpServletRequest");\nHttpServletResponse res = (HttpServletResponse)\ncontext.getValue("HttpServletResponse");\n``` The information returned from the response can be set in cookies. Note these objects only work for URL resources. This code returns null for all other types of WebLogic resources.
CR083125	Fixed a problem that caused the weblogic.security.service.URLResource class to behave differently on Windows than it did on other platforms. On Windows, this class converted the contextPath and URI to lower case. This happened because the contextPath and URI were case insensitive in Windows. To change the default behavior, you had to set one of the following flags: • -Dweblogic.security.URLResourceCaseMapping=on • -Dweblogic.security.URLResourceCaseMapping=off • -Dweblogic.security.URLResourceCaseMapping=os If you set the flag to on, the contextPath and URI always mapped to lower case. If you set the flag to off, the contextPath and URI never map to lower case. If you set the flag to os, the contextPath and URI mapped to lower case on Windows. You would only have needed to set this flag when you were running a domain on a combination of Windows and non-Windows machines. In this case, setting the flag to on corrected the problem across platforms, although it made non-Windows systems case insensitive.

Servlets and JSPs

Change Request Number	Description
CR042655	Fixed error in the time format in the extended log format of the access.log file. Time now displays in GMT.
CR074096	Fixed a problem that resulted in no Mbean access in the servlet's `<init>` method as it is executed as an anonymous user. Two new elements were added to the `weblogic.xml` deployment descriptor file to be used instead of `run-as` identify.
CR074843	Upgraded WebLogic Server so that JSP request time attributes used to output an empty string when the expression was null now output the string null.
	In previous versions of WebLogic Server, JSP request time attributes (`<%= expr %>`) output an empty string when the expression was null. Now it outputs the string "null". The JSP specification mandates that the expressions default must be "null."
	However, a new flag was introduced in `weblogic.xml` called `printNulls` which is `true` by default meaning that "null" will be the default output. Setting the flag to `false` makes sure that expressions with "null" results are printed as "", which was the earlier behavior.
	Configuring the `printNulls` tag in `weblogic.xml`:
	``` <weblogic-web-app> <jsp-descriptor> <jsp-param> <param-name>printNulls</param-name> <param-value>false</param-value> </jsp-param> </jsp-descriptor> </weblogic-web-app> ```
	Also, you can compile from `weblogic.jspc` with:
	`-noPrintNulls`
	This will also show "null" in jsp expressions as "" (an empty string).
CR075805	Implemented a way to set the current Subject of an HttpSession by adding a new `runAs` method to `weblogic.servlet.security.ServletAuthentication`.
CR076330	Fixed a problem in which the server was unable to load a Web Application class from a JAR file which had a period in its name.

Change Request Number	Description
CR077823	Updated the plug-in information in the Administration Guide and the using plug-ins Guide to reflect updated plug-in parameter information.
CR077944	Fixed a problem with JSP tag where JSP2XmlOutputter calls dequote() incorrectly which causes parse errors.
CR078698	Fixed a problem that prevented the access.log file from rotating by size when the size exceeded 2 MB.
CR078725	Fixed a problem where the HttpClusterServlet decoded the chunk transfer encoded response.
CR079892	Fixed a problem that caused CPU usage to go up to 100% while executing CGI scripts from multiple browser windows and than in the middle of that process, hitting the stop button on all browsers by preventing hung thread when executing hugh CGI scripts with charsets defined in t Context-Type.
CR079924	Provided the ability to define dispatch policy at the web-app level by adding new element, `wl-dispatch-policy`, to the `weblogic.xml` file.
CR080090	Enabled FrontEndHTTPPort, FrontEndHTTPSPort, and FrontEndPort to take universal precedence if defined. They should always be used when redirecting.
CR080445	Fixed a problem that caused WebLogic Server not to recognize WeblogicPluginEnabled=true in the server section of the config.xml file. This caused the incorrect client machine address to be passed in the header.
CR080554	Fixed a problem where IIS and iPlanet do not pass the jsessionid identifier to WebLogic Server for rewritten URLs by introducing the EncodeSessionIdInQueryParams property.
CR080608	Fixed a problem that caused `weblogic.servlet.security.ServletAuthentication` weak string to always return `FAILED_AUTHENTICATION`.
CR080613	Updated WebLogic Server to allow special characters in quoted string values; previously commas in a cookie's quoted value did not work.
CR080648	Added a new attribute called global-role to mark a given role as a global role.

Change Request Number	Description
CR080751	Fixed a problem with mapping error pages to custom exceptions. This problem occurred because custom exception class extended from ServletException was not fully compliant with the Servlet 2.3 specification.
CR080772	Fixed a problem that caused `UnsupportedEncodingExecption` if user had extra quotes around the charset.
CR080791	Provide the ability to handle folded headers properly in the HttpClusterServlet and the HttpProxyServlet.
CR081253	Fixed a problem that caused WebLogic Serve not to properly parse Http chunk encoded requests.
CR081484	Fixed a problem where session attributes were lost when using HttpSessionListener with the PersistentStoreType specified as replicated.
CR081521	Fixed a problem that caused CGI scripts, called from a JSP, not to work in a WAR or EAR file.
CR081792	Relaxed type checking for HttpServletRequestWrapper so that the servlet engine can use ServletRequestWrapper instead of HttpServletRequestWrapper.
CR082238	Added a session descriptor called `CacheSessionCookie` to the `weblogic.xml` file. This descriptor is set to `false` by default. If set to true, the `Cache-Control` header is not added.
CR082310	Fixed a problem that prevented users from deleting or overwriting JSPs, with a "page contentType" directive, in exploded WAR files. The solution was to ensure that the inputstream in the JSP Parser was closed after it was used.
CR082461	Fixed a problem that caused `FileServlet` not to work. Ensured that WebLogic Server does not prepend the `SERVLET_PATH` to the `PATH_INFO` used to construct the desired filename if a `docHome` parameter is specified.
CR082636	Fixed a problem that caused the empty `auth` constraint does not restrict access in `fullSecurityDelegation` mode.
CR083200	Fixed a problem that prevented WebLogic Server from filtering `nonProxyHosts` correctly. WebLogic Server now honors both `http.nonProxyHosts` and `nonProxyHosts`.

Change Request Number	Description
CR083517	Fixed a problem that caused `HttpClusterServlet` not to proxy requests to SSL even when the `SecureProxy` is enabled.
CR083654	Fixed performance problems by improving the of performance of Servlets and JSPs that use multibyte charsets as content type.

Tools

Change Request Number	Description
CR074535	Fixed a problem in `weblogic.Builder` that caused the DDinit not to recognize recursive relations.
CR074535	Fixed a problem in `weblogic.Builder` that caused the DDinit not to recognize recursive relations.
CR074570	Fixed a problem in `weblogic.Builder` that caused the DDinit to mistake cmr fields for cmp fields.
CR074620	Fixed a problem in `weblogic.Builder` that caused the CMP field type to be enabled when the Bi-directional field on the Wizard is checked even though the relationship is one-to-one.
CR074629	Fixed a problem in `weblogic.Builder` that caused an Illegal DatasSource dialog box to display when user pressed the cancel button on the Connect to Server dialog box for any browse action.
CR074689	Fixed a problem in `weblogic.Builder` that caused checking the create duplicate environments field to generate incomplete XML.
CR074764	Added the set `cache-type` field, missing from the tuning panel for stateful session beans to the Session Advanced panel.
CR074773	Fixed a problem that prevented the XMLElement generator from generating XML code for stateful session EJBs with `allow-concurrent-calls` specified.
CR074995	Fixed a problem in `weblogic.Builder` that occurred when user deleted a relations, the XML was removed but the module was not marked as modified. When the module was closed, user was not prompted to save the change.
CR075034	Fixed a problem in `weblogic.Builder` where on the EJB|Tuning|Cluster panel, adding a new Idempotent method caused a java.lang. IllegalArgumentException.
CR075521	Fixed a problem in `weblogic.Builder` online help where several searches on various topics displayed an incorrect page.

Change Request Number	Description
CR075537	Fixed the problem in `weblogic.Deployer` where servers added to a cluster in a running domain were not deployed on the new clusters in the server. If you added a server to a cluster and restarted the Managed Servers, Web applications previously targeted and deployed to the cluster were not automatically deployed on the added server.
CR075672	In `weblogic.Builder`, removed the Methods Are Idempotent checkbox because `methods-are idempotent` is deprecated. Developed a way to add idempotent methods for stateless session EJBs. Idempotent methods can now be specified on the Cluster panel.
CR075937	Fixed a problem where `weblogic.Builder` wrote illegal values for the EJB deployment descriptor, max-cache-size.
CR077171	Fixed a problem in `weblogic.Builder` where the DDinit sets the wrong version for the container-managed persistent (cmp) descriptor.
CR078747	Fixed a problem where `weblogic.Builder` cannot open the `ejb11_container_managed.jar` file.
CR081213	Fixed a problem in `weblogic.Builder` that caused a nullPointerException to occur when switching between stateless and stateful session bean checkboxes on the General tab for State beans.
CR081325	Fixed a problem in `weblogic.Builder` that caused the DDInit not to assign JNDI names for the Web application's EJB Refs and Resource Refs and the bean's Resource Refs.

WebLogic Tuxedo Connector

Change Request Number	Description
CR078774	Fixed the problem that caused tBridge session connections to become non-responsive when pulling messages from Tuxedo /Q.

Web Services

Change Request Number	Description
CR066226	Enabled WebLogic Server to provide two-way SSL authentication a RPC type Webservice.
CR069027	Fixed a problem that prevented dynamic from working with the wsdl client to invoke WebServices with complex datatypes.
CR072268	WebLogic Web services do not support XML Schemas that contain complexTypes that inherit from simple types.
CR073539	Fixed broken links on the WebServices Home page where an extra "/" was accidently added.
CR075634	Fixed a problem where the generated stub fails to handle the doc/lit with echo"Struct.
CR076677	Fixed a problem that prevented dynamic from being used with the wsdl client to invoke WebServices with complex datatypes.
CR077849	Fixed a problem that caused multi-byte strings to be corrupted by WebServices when using an English environment.
CR077855	Fixed a problem that causes a NullPointerException when trying to access a Webservice from a java client while using a URL.getContext().
CR078313	Fixed connection problem that occur intermittently when the stub client interop to.net service fails.

Change Request Number	Description
CR078442	Fixed a problem caused by accessing the home page through HTTPS for a Web service generated both with and without the protocol=https in services generation causes an authentication login dialog to display.
CR078804	Fixed the problem where the Web Services Stateless Session example uses the not public API import weblogic.ejb.GenericSessionBean.
CR079620	SOAP messages are now compliant with RFC2376.
CR079976	Fixed a problem that occurred when trying to specify an HTTP proxy server when using clientgen and other tasks by adding proxyHost and proxyPort attributes to the clientgen.
CR079984	Disabled the feature that checked for the presences of webservices.jar file in the WebServices examples.
CR080152	Fixed a problem with WebLogic Server parsing dateTime.
CR080154	Fixed a problem that caused an authentication screen to pop up when using HTTP protocol and an array of int or string in the method signature.
CR080264	Fixed a problem that caused the SOAP response not to have the required utf-8 contentType in the HTTP header.
CR080961	Obtained a fix for a bug in the XML parser, Xerces 1.3.1, in the org.xml.sax.helpers.AttributesImpl.removeAttribute() call.
CR081570	Fixed a problem that caused servicegen to fail when a set or get method is not found for properties by ignoring the Java bean properties that do not have both a getter and setter method.