



BEA WebLogic JRocket™ SDK

Release Notes

Version 8.1 Service Pack 2
December 2003

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

Installation	1
Contents of the SDK	1
Important Notes	2
Notes for Service Pack 2	3
J2SE Certification	3
Class or Method Names Using Multi-byte Characters Can Adversely Affect WebLogic JRockit	3
file.encoding for Linux Implementations Set to “EUC-JP-LINUX”	4
Preventing JRockit Crashes When Multiple Threads Use JNI GetFieldID	4
Memory Leak in lowmemposix with High Thread Turnover	4
Completed Native Threads Released More Often	4
DatagramPacket.getAddress() Returned Wrong Information	4
Problem with Stopping JRockit as an NT Service Corrected	5
Controlling JVM Process Size Growth	5
JNIString Implementation Now Consistent with Sun Implementation	5
64-bit JRockit Crashing with Orphaned Threads	5
Compliance Test Suite Recommendation	6
Out-of-Memory Errors Encountered when Running with WLS jDriver	6
Corrected Hangup in java.lang.Object.getMonitorIndexAndLock(Native Method). 6	
Trusted CAs in the JDK cacerts keyStore File Due to Expire	6
Notes for Service Pack 1	7

Relocating WebLogic JRockit RPMs Obtained from Red Hat	7
Avoiding Paging-related Performance Degradation	7
Avoiding a Crash when Using -Xgc:gencopy and a Set Nursery Size	8
Unified Format for -Xverbose:memory Output Across all Garbage Collectors.	8
Support for windows-31j Encoding on jDriver	8
Support for 64-bit Files on Linux	8
Unified Root Directory Names for Linux	8
Zip File Header Limitation Expanded	9
Buffer Overrun Problem Corrected	9
Using the Correct Version of glibc with IA32 Linux	9
Memory Footprint Appears Increased	9
Mimicking GA Behavior for SP1 Implementations	10
“Bad File Descriptor” Exception Thrown when Too Many Files are Open.	10
NullPointerException Thrown for Synchronized JNI-function Returning Double10	
Extensive Use of Runtime.exec() Might Cause Crash on IA64.	10
Running tnameserv and Console on Linux64.	10
Notes for GA	11
Dealing with Heavily Contended Locks.	11
Getting CPU Usage Information in the Management Console.	11
Using Linux on Hyper Threading-enabled Intel Processors.	11
Unknown pthread Library.	12
Invalid Command Line Flags	12
Documentation	12
Note About OS Support	13

BEA WebLogic JRockit 8.1 SDK Release Notes

Version 8.1 Service Pack 2

This document contains important details for this version of BEA WebLogic JRockit 8.1 SDK. It contains information on the following subjects:

- [Installation](#)
- [Contents of the SDK](#)
- [Important Notes](#)
- [Documentation](#)
- [Note About OS Support](#)

Installation

WebLogic JRockit 8.1 SDK is available either as a standalone application or as part of the BEA WebLogic Platform suite. For instructions on installing it as a standalone, please see [Installing WebLogic JRockit 8.1 SDK](#). For instructions on installing it as part of WebLogic Platform, please refer to [Installing WebLogic Platform](#).

Contents of the SDK

For a detailed description of what is included in this service pack of BEA WebLogic JRockit 8.1 SDK, please refer to [What's in the WebLogic JRockit 8.1 SDK?](#)

Important Notes

Before using WebLogic JRockit 8.1 SDK, you should be aware of the information discussed in this section. These important notes are:

- [Notes for Service Pack 2](#)
 - [J2SE Certification](#)
 - [Class or Method Names Using Multi-byte Characters Can Adversely Affect WebLogic JRockit](#)
 - [file.encoding for Linux Implementations Set to “EUC-JP-LINUX”](#)
 - [Preventing JRockit Crashes When Multiple Threads Use JNI GetFieldID](#)
 - [Memory Leak in lowmemposix with High Thread Turnover](#)
 - [Completed Native Threads Released More Often](#)
 - [DatagramPacket.getAddress\(\) Returned Wrong Information](#)
 - [Problem with Stopping JRockit as an NT Service Corrected](#)
 - [Controlling JVM Process Size Growth](#)
 - [JNIString Implementation Now Consistent with Sun Implementation](#)
 - [64-bit JRockit Crashing with Orphaned Threads](#)
 - [Compliance Test Suite Recommendation](#)
 - [Out-of-Memory Errors Encountered when Running with WLS jDriver](#)
 - [Corrected Hangup in java.lang.Object.getMonitorIndexAndLock\(Native Method\)](#)
 - [Trusted CAs in the JDK cacerts keyStore File Due to Expire](#)
- [Notes for Service Pack 1](#)
 - [Relocating WebLogic JRockit RPMs Obtained from Red Hat](#)
 - [Avoiding Paging-related Performance Degradation](#)
 - [Avoiding a Crash when Using -Xgc:gencopy and a Set Nursery Size](#)
 - [Unified Format for -Xverbose:memory Output Across all Garbage Collectors](#)
 - [Support for windows-31j Encoding on jDriver](#)
 - [Support for 64-bit Files on Linux](#)

- Unified Root Directory Names for Linux
- Zip File Header Limitation Expanded
- Buffer Overrun Problem Corrected
- Using the Correct Version of glibc with IA32 Linux
- Memory Footprint Appears Increased
- Mimicking GA Behavior for SP1 Implementations
- “Bad File Descriptor” Exception Thrown when Too Many Files are Open
- NullPointerException Thrown for Synchronized JNI-function Returning Double
- Extensive Use of Runtime.exec() Might Cause Crash on IA64
- Running tnameserv and Console on Linux64
- Notes for GA
 - Dealing with Heavily Contended Locks
 - Getting CPU Usage Information in the Management Console
 - Using Linux on Hyper Threading-enabled Intel Processors
 - Unknown pthread Library
 - Invalid Command Line Flags

Notes for Service Pack 2

This section describes issues resolved in this service pack and other product-related notes. The issues described here were originally identified in WebLogic JRockit 8.1 SDK Service Pack 1 and earlier.

J2SE Certification

This service pack of WebLogic JRockit 8.1 SDK is certified to run with J2SE version 1.4.1_05.

Class or Method Names Using Multi-byte Characters Can Adversely Affect WebLogic JRockit

Class or method names that include multi-byte characters can cause WebLogic JRockit to hang or crash. Under certain conditions, when internal strings are converted to multi-byte strings, the

JVM can cause a buffer overrun, which might result in it crashing or hanging up. If you encounter this situation, please contact support (support@bea.com) to obtain a patch to work around it.

file.encoding for Linux Implementations Set to “EUC-JP-LINUX”

WebLogic JRockit has remapped its `file.encoding` property for Linux implementations using the Japanese locale from `EUC-JP` to `EUC-JP-LINUX`.

Preventing JRockit Crashes When Multiple Threads Use JNI GetFieldID

In some circumstances, WebLogic JRockit would crash when using JNI's `GetFieldID()` and multiple threads called the function. It was discovered that the field cache was not thread safe. The JVM was modified so that read/writes are now guarded with spinlock. `GetFieldID()` now works even when multiple threads are constantly calling it.

Memory Leak in lowmemposix with High Thread Turnover

This version of WebLogic JRockit modifies the thread clean-up code to correct a problem with how the JVM's thread clean up resulted in their stack memory being left behind, thus, a memory leak. This was a race condition that meant a few stacks for every thousand threads cleaned up were not freed.

This problem occurred in Java programs that had a high thread turnover rate. The user would notice a gradual increase in process size over time, indicative of a memory leak. Java programs using Oracle JDBC drivers under certain conditions would also notice this problem, since this Oracle's driver may create threads for canceling transactions.

Completed Native Threads Released More Often

It was possible for an application with large heap to allocate enough threads to exceed the maximum, process size even when those threads have completed. WebLogic JRockit has been modified in this service pack to release terminated platform threads every 25 completions.

DatagramPacket.getAddress() Returned Wrong Information

When used in a Linux 32-bit implementation, the `DatagramPacket.getAddress()` was returning the first client address from which the packet has been received, whereas the Windows32-bit version would return the proper source address. WebLogic JRockit has been updated as of this version to correct this situation.

Problem with Stopping JRockit as an NT Service Corrected

Some users experienced a problem when they started WebLogic JRockit an NT server and then stopped the service. Windows would generate an error message and then advise them that the Java service would be closed. Another message would then that indicate that the service could not be stopped. This situation has been corrected.

Controlling JVM Process Size Growth

The version of WebLogic JRockit corrects a problem with the JVMs ability to control process size growth. In earlier versions of WebLogic JRockit, there were occasional problems controlling this growth, even if the Java heap was well within its limit and was almost constant during the run. This indicated a memory leak outside the heap, with the process size growing up to 2 GB before the process hung up or died.

JNIString Implementation Now Consistent with Sun Implementation

In earlier versions of WebLogic JRockit, the JNI implementation was inconsistent with Sun's implementation. This inconsistency occasionally resulted in `emptyString` and `nullString` failures. The WebLogic JRockit implementation of JNI is now consistent with Sun's.

64-bit JRockit Crashing with Orphaned Threads

This service pack of WebLogic JRockit corrects a problem wherein Java applications having high thread turnover were prone to random crashes or hanging of the JVM. This problem occurred on all platforms, but was most frequently seen on 64-bit Linux machines. If a JRockit dump was produced after a crash or hangup, it would often be incomplete or show different problems each time it was generated.

Special Note to 64-bit Linux Users

If you are using 64-bit Linux, ensure that your glibc and kernel versions are up to date with the latest bug fixes from the [RedHat Network](#):

- Kernel: 2.4.18-e.37smp (Pthread+fork fix).
- Glibc: 2.2.4-32.8.ia64; required after RedHat issued RHSA-2003:249 and contains these critical IA64 fixes:
 - Corrected a race condition that existed in the `malloc` routine for IA64 platforms, which could cause memory corruption.

- Corrected an error in `pthread_spinlocks` that prevented spinlocks from functioning correctly on IA64 platforms.

Compliance Test Suite Recommendation

If you are running the J2EE compliance test suite from Sun with WebLogic JRockit on WebLogic Server, we recommend that you set `-Xmx` and `-Xmx` to the same size to prevent the system from dumping. If you do encounter a dump on startup, review your `-Xmx` and `-Xmx` settings and reduce them, if necessary.

Out-of-Memory Errors Encountered when Running with WLS jDriver

Some users have encountered an out-of-memory error when WebLogic JRockit tried to allocate a new byte array when running with jDriver. When the allocation fails `glibc` threw a C++ exception which wasn't not caught, causing the program to dump core and terminate. At the same time, no `jrockit.dump` file is created. This problem has been corrected.

Corrected Hangup in `java.lang.Object.getMonitorIndexAndLock(Native Method)`

Occasionally, customers have seen some execute threads hang up in a `java.lang.Object.getMonitorIndexAndLock(Native Method)`, call while the CPU pegs to 100% utilization. This version of WebLogic JRockit has been enhanced to alleviate this problem by implementing the correct handling of native weak handles, which are more reachable than final and phantom handles.

Trusted CAs in the JDK cacerts keyStore File Due to Expire

The JDK cacerts file and the WebLogic Server cacerts file are keystore files that contain out-of-the-box trusted Certificate Authority certificates. These trusted CA certs are used in certificate chain verification when using SSL.

In this service pack, WebLogic Server checks each trusted CA certificate in the JDK and WebLogic Server cacerts files for the NotAfter date. If the trusted CA is due to expire (that is, exceed the NotAfter date) within 30 days, a warning message is output about this impending expiration. If a trusted CA has already expired, with one exception, a warning message is output declaring the trusted CAs as having exceeded the NotAfter date.

Two trusted CAs in the JDK cacerts keystore file are due to expire approximately at the release date of this service pack. Thus, you might see one or both of these warning messages output while booting WebLogic Server.

If you use either of these two trusted CAs in your server certificate chain, you should obtain a new trusted CA or certificate from your Certificate Authority of choice (Verisign, Certicom, and so on).

The current JDK cacerts file contains at least one certificate from Verisign that has expired (the Class 4 Certificate Authority, alias verisignclass4ca). No certificate expired message is output when this trusted CA is loaded.

For more information, see Sun Alert 57436 at

<http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F57436>.

Notes for Service Pack 1

This section describes issues resolved in this service pack and other product-related notes. The issues described here were originally identified in WebLogic JRockit 8.1 SDK GA and earlier.

Relocating WebLogic JRockit RPMs Obtained from Red Hat

While the BEA download site only distributes WebLogic JRockit packages using the BEA installer, if you are subscribed to Red Hat Network, you can obtain WebLogic JRockit binaries in RPM format from the Red Hat Enterprise Linux AS channel. If you do obtain the WebLogic JRockit RPMs, you can relocate them from the default directory (`/opt/bean/jrockit81_141`) to a directory you specify.

For complete details, please refer to “[Overriding the Default Installation Path with RPM](#)” in *Installing WebLogic JRockit*.

Avoiding Paging-related Performance Degradation

If you don't set the maximum heap size (`-Xmx`), fragmentation might cause paging to occur, degrading system performance. This is because WebLogic JRockit grows the heap aggressively when fragmentation occurs, potentially out-stripping the physical memory available. Although WebLogic JRockit sets the default maximum heap size to 75% of physical memory in the system, it will grow the heap aggressively when the heap becomes fragmented. To avoid this situation, you should override the default and manually set `-Xmx` to no more than 75% of the available physical memory.

For complete details, please refer to “[Setting the Initial and Maximum Heap Size](#)” in *Tuning the WebLogic JRockit JVM*.

Avoiding a Crash when Using `-Xgc:gencopy` and a Set Nursery Size

Starting WebLogic JRockit with the generational copy garbage collector (`-Xgc:gencopy`) and a nursery explicitly set to more than 15% of initial heap size will cause the JVM to crash. To avoid this situation, either accept the default nursery size by *not* specifying `-Xns` or set the nursery size to less than 15% of the `-xms` setting.

For more information on setting the nursery size, please refer to [“Setting the Size of the Nursery”](#) in [Tuning the WebLogic JRockit JVM](#).

Unified Format for `-Xverbose:memory` Output Across all Garbage Collectors

To ensure consistency of information displayed when `-Xverbose:memory` is specified, the output for that option has been unified across all garbage collectors; that is, specifying `-Xverbose:memory` will result in the same type of information displayed, regardless of the garbage collector used. For details, please refer to [“Displaying Logging Information”](#) in [Starting and Configuring WebLogic JRockit](#).

Support for windows-31j Encoding on jDriver

This version of WebLogic JRockit (J2SE 1.4.1) now supports MS932 windows-31j encoding on jDriver. Previously, this support was not available because the windows-31j converter was not included in `lib/charsets.jar`. This affected WebLogic JRockit’s support for Japanese language implementations.

Support for 64-bit Files on Linux

When running WebLogic JRockit on Linux some users were experiencing a “File size limit exceeded” exception and the application stopped, indicating that the JVM was not able to handle large (greater than 2 GB) files. This issue was reported in CR103219 and has been corrected. WebLogic JRockit now supports 64-bit files.

Unified Root Directory Names for Linux

The root directory names for WebLogic JRockit SDK Linux installations are now the same, regardless of the Intel architecture used. Based on the installation method you use, the root directories are as listed in [Table 1](#):

Table 1 Unified Root Directory Names

Installer	Root Directory for IA32	Root Directory for IA64
RPM	/opt/bea/jrockit81sp1_141_03/	/opt/bea/jrockit81sp1_141/
BEA	\$BEAHOME/jrockit81sp1_141_03/	\$BEAHOME/jrockit81sp1_141/

Note: When available, a microversion number will be added to the directory name for IA64.

Zip File Header Limitation Expanded

Although the JDK allows for zip file headers to contain more than 256 bytes of additional information, an exception was being thrown when these files became larger than that. This issue was reported in CR104622. In response, the `zipFile()` constructor was modified to support up to 64 KB of extra header information.

Buffer Overrun Problem Corrected

A confluence of coding and the compiler used on that code was causing a rare buffer overrun (see CR099551). This situation has been corrected and future problems of this sort should no longer occur.

Using the Correct Version of glibc with IA32 Linux

Red Hat Enterprise Linux AS / ES / WS 2.1 users need to ensure that `glibc` version `i686` is installed, not `glibc` version `i386` version. Otherwise, you might encounter a segfault error upon startup. Additionally, do not add the folder `/lib` or `/lib/i386` to your `LD_LIBRARY_PATH`. If `/lib` is in `LD_LIBRARY_PATH`, `/lib/libpthread.so` will be the linked runtime, instead of `/lib/i686/libpthread.so`.

Memory Footprint Appears Increased

The implementation of `java.lang.Runtime.freeMemory()` has been corrected. Previously, it regarded the whole nursery as being free. Because of this, an application may incorrectly get the impression that WebLogic JRockit now uses more memory. This is not the case since previously a higher value than was actually free was returned.

Mimicking GA Behavior for SP1 Implementations

If you start WebLogic JRockit with the `gencon` garbage collector (either implicitly, as the default, or explicitly) and you specify explicit values for `-Xmx` and `-Xns`, you will need to double the value specified for `-Xns` to mimic the GA behavior.

“Bad File Descriptor” Exception Thrown when Too Many Files are Open

CR101146 reports that WebLogic JRockit might throw an `IOException: Bad file descriptor` if too many files are open. To work around this situation, increase the maximum number of files you can open to 2048 by setting `ulimit -n 2048`. You can only increase this number if the maximum number of files you can open is equal to or greater than 2048. If maximum number of processes is lower than 2048, have your system administrator increase it to at least 2048.

NullPointerException Thrown for Synchronized JNI-function Returning Double

As reported in CR107067, a JNI function declared as:

```
public native synchronized float foo();
```

will throw a `NullPointerException` upon returning to the JNI-stub. This issue will be resolved in WebLogic JRockit 8.1 Service Pack 2.

Extensive Use of Runtime.exec() Might Cause Crash on IA64

If you are running a Linux version of WebLogic JRockit on an IA64 SMP machine and are using `Runtime.exec()` extensively, you might encounter arbitrary crashes during processing. This issue, reported in CR104064, is caused by an O/S-level problem that has been resolved by a patch now available from RedHat at:

<http://rhn.redhat.com/errata/RHSA-2003-198.html>

Running tnameserv and Console on Linux64

As reported in CR111919, if you are a Linux64 user, you will encounter a problem running `tnameserv` and the JRockit Management Console. You will see one of these error messages:

```
Error: could not find libjava.so
```

```
Error: could not find Java 2 Runtime Environment
```

To workaroud this situation, from the command line, enter:

```
cd <PATH_TO_JROCKIT_81SP1>/jre/lib
ln -s ia64 sparcv9
```

Once you done this, you should be able to run `tnamserv` and the Console by using your normal scripts.

Notes for GA

This section describes the issues were resolved in WebLogic JRockit 8.1 SDK GA and other important product-related information. The issues described here were originally identified in WebLogic JRockit 8.0 SDK and earlier.

Dealing with Heavily Contended Locks

If you are running a high-stress application with heavily contended locks on a multiprocessor system, you can attempt to improve performance by using the option `-XXenablefatspin`. This option enables the ability to spin for a short time before going to sleep. While in most cases, `-XXenablefatspin` improves performance slightly—particularly if the contended locks in your code are only held for a short period of time—be aware that, in some cases (perhaps up to 20%), it degrades performance. Future versions of WebLogic JRockit are expected to automatically adjust the spinning for different locks, eliminating the need for this option.

Getting CPU Usage Information in the Management Console

To be able to get CPU usage information in the JRockit Management Console from a JRockit instance running on Windows 2003 Server, the JRockit instance needs to be running as a user who has administrative rights on the computer in question.

Using Linux on Hyper Threading-enabled Intel Processors

If you are using Linux on Hyper Threading-enabled Intel processors, be aware of the following:

- To make sure that JRockit can detect that Hyper Threading is enabled you need to make the devices in `/dev/cpu/X/cpuid` (where `X` is the `cpu` number) readable for any process. This is not a security problem, since almost all of this information is available through `/proc/cpuinfo` anyway, and it is read only.
- For the device to be loaded automatically you also need to add the line `alias char-major-203 cpuid` to your `/etc/modules.conf` file.

- JRockit will still work on an Hyper Threading enabled system even if it doesn't detect that Hyper Threading is enabled, although Hyper Threading-specific optimizations will not be performed in this case.

Unknown pthread Library

If you receive the following error:

```
ERROR: The pthread library is unknown. Are you running a supported Linux distribution? (Segment-register gs appears to be unused)
```

You should make sure that the environment variable `LD_ASSUME_KERNEL=2.2.5` is *not* set.

Invalid Command Line Flags

Be sure to use only valid command line options when starting WebLogic JRockit JVM. If you use an invalid option, WebLogic JRockit will exit. Note that this behavior is different from that of previous versions of the VM, where it was ignored. For a list of valid command line options, please refer to [Command Line Options by Name](#), in *Tuning WebLogic JRockit JVM*.

Documentation

The documentation set has been completely updated and reorganized since WebLogic JRockit 8.0. Significant changes include the following:

- A new tuning guide, *Tuning WebLogic JRockit JVM*, is included with this version of WebLogic JRockit SDK.
- Instructions for migrating to WebLogic JRockit 8.1 JVM from other JVMs, particularly Sun Microsystems' HotSpot JVM, are now available in the *Migration Guide*. This document also includes best coding practices for J2SE applications, a troubleshooting guide, and instructions for using JVMPI and JVMDI.
- Information about using WebLogic JRockit [with BEA WebLogic Server](#)—including optimal configuration parameters and procedures for monitoring JRockit from the WebLogic Server Administration Console—has been added to the *Using WebLogic JRockit*.
- New installation information has been added to *Installing WebLogic JRockit 8.1*.

Copies of all WebLogic JRockit 8.1 SDK documents can be found at:

<http://edocs.bea.com/wljrockit/docs81/index.html>

Note About OS Support

BEA will support WebLogic JRockit 8.1 SDK and provide bug fixes on only those operating system distributions that have been certified by BEA. While it may be possible to run WebLogic JRockit on other non-certified distributions, BEA makes no such guarantees. In addition, BEA will not be responsible for providing any bug fixes for non-certified distributions.

