



BEA JRockit JDK R26

Release Notes

Release R26.4
July 2006

Copyright

Copyright © 1995-2006 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS 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 DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks and Service Marks

Copyright © 1995-2006 BEA Systems, Inc. All Rights Reserved. BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Interaction, BEA AquaLogic Interaction Analytics, BEA AquaLogic Interaction Collaboration, BEA AquaLogic Interaction Content Services, BEA AquaLogic Interaction Data Services, BEA AquaLogic Interaction Integration Services, BEA AquaLogic Interaction Process, BEA AquaLogic Interaction Publisher, BEA AquaLogic Interaction Studio, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Kodo, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA SALT, BEA Service Architecture Leveraging Tuxedo, BEA WebLogic Commerce Server, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Mobility Server, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Real Time, BEA WebLogic RFID Compliance Express, BEA WebLogic RFID Edge Server, BEA WebLogic RFID Enterprise Server, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, BEA Workshop for WebLogic Platform, BEA Workshop JSP, BEA Workshop JSP Editor, BEA Workshop Struts, BEA Workshop Studio, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. Accelerated Knowledge Transfer, AKT, BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

Contents

BEA JRockit R26 JDK Release Notes

License Agreement	2
Platform Support	2
Java Support	2
Installation	2
Documentation Accompanying the BEA JRockit Release	2
New Features and Enhancements in BEA JRockit R26.4	3
Performance Improvements on 64-bit Platforms	3
Performance Improvements for Low Latency Applications	4
Performance Improvements on SPARC	5
Additional Tuning Possibilities	5
New Features and Enhancements in BEA JRockit R26.3	6
New Features and Enhancements in BEA JRockit R26.2	6
New Features and Enhancements in BEA JRockit R26.1	6
New Features and Enhancements in BEA JRockit R26.0	6
Most Recent Changes	6
Changes in the BEA JRockit R26.4 Release	7
Changes in the BEA JRockit R26.3 Release	9
Changes in the BEA JRockit R26.2 Release	12
Changes in the BEA JRockit R26.0 Release	14
Known Issues	18

BEA JRockit R26 JDK Release Notes

Version R26 JDK

This document contains important details for BEA JRockit R26 JDK. It contains information on the following subjects:

- [License Agreement](#)
- [Platform Support](#)
- [Java Support](#)
- [Installation](#)
- [Documentation Accompanying the BEA JRockit Release](#)
- [New Features and Enhancements in BEA JRockit R26.4](#)
- [New Features and Enhancements in BEA JRockit R26.3](#)
- [New Features and Enhancements in BEA JRockit R26.2](#)
- [New Features and Enhancements in BEA JRockit R26.1](#)
- [New Features and Enhancements in BEA JRockit R26.0](#)
- [Changes in the BEA JRockit R26.4 Release](#)
- [Changes in the BEA JRockit R26.3 Release](#)
- [Changes in the BEA JRockit R26.2 Release](#)

- [Changes in the BEA JRockit R26.0 Release](#)
- [Known Issues](#)

License Agreement

The BEA JRockit R26 JDK is subject to the terms and conditions of the BEA JRockit Binary License Agreement, see the [BEA License Agreement](#).

Platform Support

- The BEA JRockit 5.0 (R26) JDK release is certified on the platforms listed on the following [Supported Configurations](#) page.
- The BEA JRockit 1.4.2 (R26) JDK release is certified on the platforms listed on the following [Supported Configurations](#) page.

Java Support

- The BEA JRockit R26.0 JDK release is certified to be compatible with J2SE 5.0 update 4.
- The BEA JRockit R26.1 JDK release is certified to be compatible with J2SE 5.0 update 4.
- The BEA JRockit R26.2 JDK release is certified to be compatible with J2SE 1.4.2_10.
- The BEA JRockit R26.3 JDK release is certified to be compatible with J2SE 1.4.2_10 and J2SE 5.0 update 6.
- The BEA JRockit R26.4 JDK release is certified to be compatible with J2SE 1.4.2_11 and J2SE 5.0 update 6.

Installation

BEA JRockit JDK is available as a standalone application. For instructions on installing BEA JRockit, please see [Installing BEA JRockit JDK](#).

Documentation Accompanying the BEA JRockit Release

The documentation that is connected to a specific release of BEA JRockit is located here:

<http://edocs.bea.com/jrockit/releases/index.html>

New Features and Enhancements in BEA JRockit R26.4

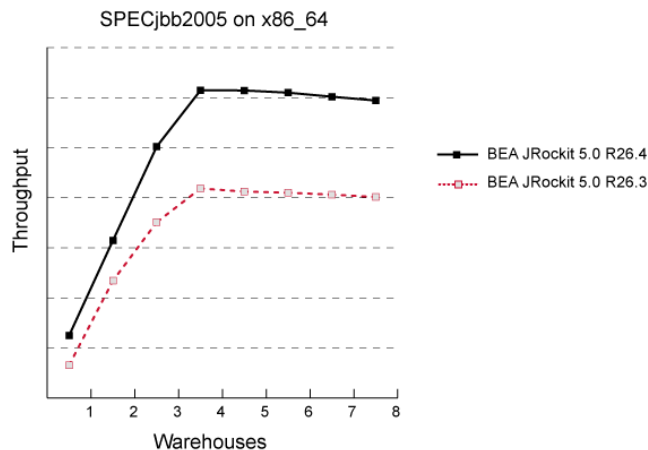
BEA JRockit R26.4 provides, apart from full support for J2SE 5.0 on all supported platforms, the following areas of improvement:

- [Performance Improvements on 64-bit Platforms](#)
- [Performance Improvements for Low Latency Applications](#)
- [Performance Improvements on SPARC](#)
- [Additional Tuning Possibilities](#)

Performance Improvements on 64-bit Platforms

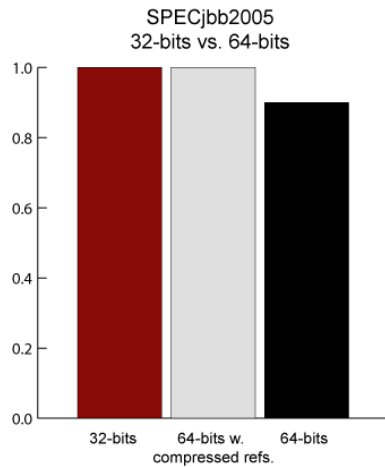
- General improvements on x86_64 platforms. See [Figure 1](#) for a comparison between the BEA JRockit R26.3 and R26.4 releases.

Figure 1 Comparison between BEA JRockit R26.3 and R26.4



- Compressed references has greatly increased the performance on 64-bit platforms. See [Figure 2](#) for a comparison of a 64-bit platform with and without compressed references compared to a 32-bit platform. For more information on compressed references, see `-XXcompressedRefs` in the [BEA JRockit Reference Manual](#).

Figure 2 Comparison between 64-bit platform with and without compressed references and a 32-bit platform.



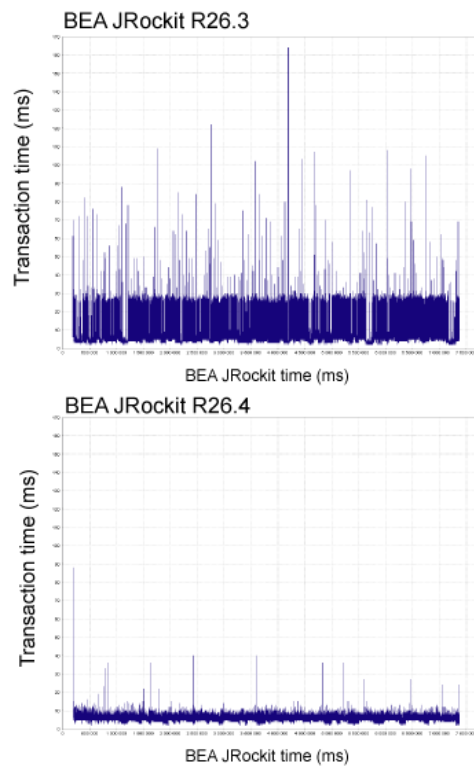
Performance Improvements for Low Latency Applications

The performance of BEA JRockit has gone through major improvements for low latency applications by the following additions:

- Lock tuning on all supported platforms—Intel Itanium, Intel Xeon, AMD Opteron, Sun SPARC (including T1), and Fujitsu SPARC 64 V.
- Incremental handling of weak native handles have been moved to the concurrent phase, which lowers the pause times.

See [Figure 3](#) for an illustration of how the performance have been improved with the above improvements.

Figure 3 Performance improvements compared between BEA JRockit R26.3 and R26.4



Performance Improvements on SPARC

SPARC performance running BEA WebLogic Server has been improved by approximately 15% compared to BEA JRockit R26.3.

Additional Tuning Possibilities

The following tuning options have been added in this release:

- [XXgcThreads](#)
- [XXoptThreads](#)
- [XXcompressedRefs](#)
- [XXlazyUnlocking](#)
- [XXthroughputCompaction](#).

For complete information on how to use all tuning options in BEA JRockit, please see the [BEA JRockit Reference Manual](#).

New Features and Enhancements in BEA JRockit R26.3

- There have been performance enhancements to the Deterministic garbage collector:
 - More efficient compaction.
 - Improved handling of reference objects.

New Features and Enhancements in BEA JRockit R26.2

- This is the first BEA JRockit JDK 1.4.2 release containing full support for the Memory Leak tool and the JRA.

New Features and Enhancements in BEA JRockit R26.1

- BEA JRockit is now available for Solaris on Sparc. Please see [BEA JRockit Supported Configurations](#) for exact configurations.

Note: The Sparc version of BEA JRockit is a 64-bit JVM only.

New Features and Enhancements in BEA JRockit R26.0

- Significant improvements in code generation speed, especially for large methods, e.g. jsp:s.
- Deterministic garbage collector (requires a separate license).
For more information, see the [Memory Management Guide](#).

Most Recent Changes

This has happened with BEA JRockit:

- [Changes in the BEA JRockit R26.4 Release](#)
- [Changes in the BEA JRockit R26.3 Release](#)
- [Changes in the BEA JRockit R26.2 Release](#)
- [Changes in the BEA JRockit R26.0 Release](#)

Changes in the BEA JRockit R26.4 Release

The following CRs have been corrected for the BEA JRockit R26.4 release.

Change Request ID	Description
CR279188	<p>When JRockit 1.4.2_10 R26.3 calculates the SUID for a class to be serialized it includes the synthetic bits in the calculation, which generates a SUID that differs from the Sun 1.4.2_10 and BEA JRockit 1.4.2_08 R24. This problem causes serialization to fail.</p> <p>Synthetic attributes are included in a class when the class references a class pointer <code>ATestClass.class</code> or uses assertions.</p> <p>This means that RMI communication between 1.4.2_10 R26.3 and Sun 1.4.2_10 can fail. RMI communication between BEA JRockit 1.4.2_10 R26.3 and BEA JRockit 1.4.2_08 R24.5 can also fail in the same way. Serialized classes, stored in a database, might also fail to be loaded.</p> <p>Note: This has been fixed in R26.4.</p>
CR258122, CR258159, CR264680	<p>The following new options have been added:</p> <ul style="list-style-type: none"> • <code>-XXgcTreads</code> • <code>-XXlazyUnlocking</code> • <code>-XXthroughputCompaction</code> • <code>-XXinternalCompactionMultiplier</code> • <code>-XXexternalCompactionMultiplier</code> <p>For a description of these options, please see the BEA JRockit Reference Manual.</p>
CR258934	<p>New arguments have been added to enable control of large pages (<code>-XXlargePages</code>) on Solaris.</p> <p>For a description of <code>-XXlargePages</code>, please see the BEA JRockit Reference Manual.</p>
CR267987	<p>Using <code>java.nio.channels.SelectionKey.OP_CONNECT</code> will make BEA JRockit block forever. This has now been fixed.</p>
CR268133	<p>Previously <code>java.lang.reflect.Method.getParameterAnnotations()</code> returned the wrong result for methods that did not have annotations. This has now been fixed.</p>
CR268439	<p>Previously when calling JVM TI functions from a non-attached thread caused BEA JRockit to crash. This has now been fixed.</p>
CR269375	<p>For each old collection, the reason for a garbage collection is printed in the <code>verbose:memdbg</code>.</p>

Change Request ID	Description
CR271551	Previously buffers that had been allocated through <code>ByteBuffer.allocateDirect()</code> would not be released, even when discarded by the application, which caused C heap memory leaks. This has now been fixed.
CR272364	The JVMTI function <code>GetThreadInfo</code> returned an error if passed NULL as the thread. It now treats NULL as the current thread in accordance with the specification.
CR274636	Previously BEA JRockit 1.4.2 R26.2 and R26.3 threw a <code>java.lang.NullPointerException</code> when passing a null argument to <code>java.lang.String.getBytes(String charsetName)</code> . This was not an issue with BEA JRockit 1.4.2 R24. This has now been fixed.
CR275108	Previously, when using the <code>lookfor</code> parameter to the Ctrl-Break handler <code>print_object_summary</code> could crash BEA JRockit if certain kinds of references were found. This has now been fixed. In addition, the output includes the kind of each reference.
CR275725	The implementation of BEA JRockit's <code>System.nanoTime()</code> on Linux provided the time since BEA JRockit started. Now BEA JRockit uses <code>gettimeofday</code> instead, which makes it easier to use the time to compare different JVM instances.
CR276308	The verbose framework in BEA JRockit is now stricter and possible ambiguous shortcuts have been removed, for example, the option <code>-Xverbose:compact</code> no longer works, instead you need to use <code>-Xverbose:compaction</code> .
CR276311	BEA JRockit R26.3.0 could, under rare circumstances, crash when using <code>-Xgc:gencon</code> . All platforms were affected. This has now been fixed.
CR276460	BEA JRockit no longer fails on RHEL4 when using <code>FileChannel.transferTo</code> .
CR276987	In BEA JRockit R26 releases prior to R26.4, a breakpoint could be garbage collected if the garbage collection went off before the method was generated. This has now been fixed.
CR277702	The <code>JraRecordingStarter.jar</code> is no longer shipped with BEA JRockit. Use <code>jrcmd</code> instead to start a JRA recording.
CR278411	Previously, BEA JRockit incorrectly threw a <code>BufferUnderflowException</code> in <code>java.nio.DirectByteBuffer.put()</code> instead of <code>BufferOverflowException</code> when the buffer overflowed. This has now been fixed.

Change Request ID	Description
CR279712	Previously, the Ctrl-Break handler <code>print_threads</code> could result in a memory leak. This was due to handles that were not released during the printing of the thread status. This problem has been noted on Linux and Solaris platforms only and it has now been fixed.
CR280443	Previously when you expanded types in the Type Graph in the BEA JRockit Memory Leak Tool, BEA JRockit hung and became unresponsive while consuming CPU resources. This issue has now been fixed.

Changes in the BEA JRockit R26.3 Release

The following CRs have been corrected for the BEA JRockit R26.3 release.

Change Request ID	Description
CR257687	In BEA JRockit R26 releases prior to R26.3, BEA JRockit could crash due to a bug in the implementation of <code>Class.getMethod()</code> . The “this” reference was not treated properly by the garbage collector if threads were stopped in bad locations. This has now been fixed.
CR258200	In BEA JRockit R26 releases prior to R26.3, BEA JRockit running could livelock due to a thread priority issue on Windows platforms. This has now been fixed.
CR179421	BEA JRockit no longer misses to report some contended monitors to JVMPI.
CR189181	Explicit font support for Asianux, part of Red Flag Linux, has been added for BEA JRockit 5.0. The font support is based on the Red Hat Linux font support.
CR247026	In earlier BEA JRockit R26 versions, on Windows operating systems, BEA JRockit could sometimes expose a problem in the OS related to multimedia timers that caused the system time to be adjusted backwards. This could cause the system time to jump back by about 1 minute. If this happened, you could turn off the use of multimedia timers with <code>-Djrockit.periodictask.usemmtimers=false</code> . This problem has now been fixed.

Change Request ID	Description
CR248132	Explicit font support for Asianux, part of Red Flag Linux, has been added for BEA JRockit 1.4.2, keeping the support for Red Flag 4.1 font patch CR200703 for LANG=zh_CN.GB2312 locale. The font support is based on the Red Hat Linux font support.
CR251838	Previously a problem regarding the handling of TrueType fonts for certain font files caused a call to <code>java.awt.Font.getXXX()</code> methods that resulted in an <code>IllegalArgumentException</code> being thrown. This issue has now been fixed. (See, CR252610 for more information).
CR252610	The problem regarding the handling of TrueType fonts for certain font files has now been fixed.
CR254297	A race condition existed in BEA JRockit when the Memory Leak Server was stopped immediately after a client had connected, which caused sockets to be left in <code>CLOSE_WAIT</code> state. On Linux, this could eventually cause the BEA JRockit process to run out of file descriptors. This problem has now been fixed.
CR255271	<ul style="list-style-type: none"> The compaction for the deterministic garbage collector has been improved. The following options have been added: <ul style="list-style-type: none"> <code>-XXinitialPointerVectorSize:<nn></code> <code>-XXmaxPooledPointerVectorSize:<nn></code> <code>-XXpointerMatrixLinearSeekDistance:<nn></code> The option <code>-XXcompactSetLimit</code> now also works with <code>-Xgcprio:deterministic</code>. <p>For more information on how to use the options, see Reference Manual.</p>
CR255959	A fix for SUN bug 5103041 has been added.
CR256867	The initialization of the initial (main) thread on Linux now respects the <code>-Xss:<size></code> option, and commits an area corresponding to the largest of this and the current system stack rlimit (see ‘ <code>man rlimit</code> ’).
CR257039	<p>The new option <code>-XXdisableGCHeuristics</code> disables the dynamic garbage collector selection heuristics when <code>-Xgcprio</code> is used. The JMAPI for changing garbage collectors in runtime can still be used.</p> <p>Note: If you disable the dynamic garbage collector selection heuristics, it will affect the behavior of the dynamic garbage collector and may lead to lower throughput or longer garbage collection pauses.</p>

Change Request ID	Description
CR257799	JRockit now allows more than one JVMTI agent to get the <code>can_tag_objects</code> capability.
CR257905	(Linux only) When BEA JRockit was started from a process that blocks signal (such as Sun's HotSpot JVM) it was not possible to send SIGQUIT to or use the <code>jrcmd</code> tool against BEA JRockit. This has now been fixed by unblocking signals that BEA JRockit listens to.
CR258248	You can now use the environment variable <code>JROCKIT_DUMP_PATH</code> to tell BEA JRockit to save crash dump information in a different location than the current working directory. The path specified must exist and be writable.
CR258395	BEA JRockit never called the JVMTI function <code>Agent_OnUnload</code> when the JVM shut down. This has now been fixed.
CR258894	When running BEA JRockit with <code>gcprio:paustime</code> or <code>gcprio:deterministic</code> , it is now possible to set the pause time target using JMAPI or JLMEXT. See the javadocs for further information.
CR259231	Previously the JVMTI function <code>GetObjectsWithTags</code> was broken. This has now been fixed.
CR260241	BEA JRockit sometimes crashed while stepping into an <code>ArrayIndexOutOfBoundsException</code> in the debugger on x86. This has now been fixed.
CR260490	The new version of <code>rtmon (librtm.so)</code> is now compatible with Montecito systems, even if the kernel is not patched with the Montecito perfmon-2 patch. This means that there are no more failing on calls to <code>RTMonRegisterThread()</code> or <code>RTMonStart()</code> , which earlier caused BEA JRockit create a core dump.
CR262448	Multiple issues with <code>-xdebug</code> in R24.5.0 have been fixed in this release.
CR262527	JRockit no longer fails to handle bytecode where the control flow enters an exception handler without a thrown exception.
CR262540	Previously when running HP OpenView Java Diagnostics Profiling Agent could cause crashes with BEA JRockit R26.0.0. This issue has now been fixed.
CR262571	Previously the <code>sizeof</code> parameter to the <code>Ctrl-Break</code> handler <code>print_object_summary</code> did not work as expected. This has been fixed.

Change Request ID	Description
CR262962	Hyperthreading detection has changed slightly, which makes BEA JRockit better at detecting whether Hyperthreading is available and turned on or off. This has also caused the BEA JRockit property <code>jrockit.cpu.ia32.ht</code> to default to the value “os” (OS detection) instead of “hw” (Hardware detection).
CR263376	RHEL 4.0 QU1 on Itanium contains a critical kernel bug that was corrected in QU2. Therefore, BEA require that you run BEA JRockit with RHEL 4.0 QU2 (or later) on Itanium systems. For IA32 and x64, there are no known issues that require an update to QU2, and you can stay on QU1.
CR264064	The default stack size for Solaris on Sparc has doubled to 512 kB.
CR265225	The BEA JRockit JRE installer in Silent mode will now be correctly installed if the <code>USER_INSTALL_DIR</code> in the XML-file has been set. See CR265227 for the previous known issue.

Changes in the BEA JRockit R26.2 Release

The following CRs have been corrected for the BEA JRockit R26.2 release.

Change Request ID	Description
CR256719	The JVM experiences a slowdown with large number of threads doing reflective invocation of the same method concurrently.
CR239984	In earlier releases of the BEA JRockit JDK 1.4.2, it was necessary to specify <code>-Xmanagement:com=com.JRockit.management.rmp.RmpSocketListener</code> on the command line to start up the Management Server. This can now be done by simply specifying <code>-Xmanagement</code> or by setting the management server port with <code>jrockit.managementserver.port</code> .
CR249272	The property <code>jrockit.managementserver.usejmx</code> has been added for BEA JRockit JDK 1.5. Setting this property to false will make BEA JRockit use the RMP-protocol instead of the default management protocol (JMX) on a BEA JRockit JDK 1.5.
CR250218	When trying to print heap references, non-fatal JVMPI error messages were displayed. This has now been fixed.

Change Request ID	Description
CR250712	A race could cause BEA JRockit to crash during a JRA recording if a thread completed at the wrong moment. This race has been fixed.
CR252315	The compaction heuristics now ignore exceptional compactions when adjusting the compact ratio and pointerset limit.
CR252348	The option <code>-Xverbose:cpuinfo</code> is now available on IA64.
CR252567	The default stack size on X64 platforms has doubled from previous releases.
CR253588	The amount of free space is now calculated correctly when BEA JRockit calculates the maximum allowed nursery size during automatic nursery resizing.
CR253952	Several JVMPI problems have been fixed.
CR254354	For some garbage collectors, the minimum block size value, set by <code>-XXminBlockSize</code> , was also (incorrectly) used as thread-local area size. With this fix, increasing the <code>-XXminBlockSize</code> value will no longer affect the thread-local area size. If you have been using <code>-XXminBlockSize</code> to adjust the thread-local area size, you now must also set <code>-XXlargeObjectLimit</code> and <code>-XXtlaSize</code> to the same value as you set <code>-XXminBlockSize</code> , as described in the BEA JRockit Reference Manual .
CR257184, CR257379	The option <code>-Xpausetarget</code> didn't always work when running <code>-Xgcprio:deterministic</code> . This has now been fixed.
CR257540	The JNI function <code>GetDirectBufferAddress</code> has now been changed to work with all direct <code>java.nio Buffers</code> . Previously it only worked for direct <code>java.nio ByteBuffer</code> s.
CR257840	To make it easier to diagnose JRockit crashes, the option <code>-XXdumpfullstate</code> has been made default. This means that if BEA JRockit crashes a lot more information is saved to disk than was the case previously. To get the old behavior use <code>-XXdumpsize:normal</code> .
CR258002	Loading extra data from zip-file entries now work.
N/A	Java Web Start and the Browser Java Plugin was included in the previous 1.4.2 BEA JRockit version, i.e. BEA JRockit 1.4.2_08 R24.5.0. These features are dropped for BEA JRockit R26 on 1.4.2.

Changes in the BEA JRockit R26.0 Release

The following CRs have been corrected for the BEA JRockit R26.0 release.

Change Request ID	Description
CR252874	In BEA JRockit 5.0 R25.2, there was an issue where an unexpected socket exception could occur, throwing the message <code>java.net.SocketException: Socket Closed:Descriptor not a socket</code> . This problem has been fixed.
CR211951	In previous versions of BEA JRockit, the JVM Process Load was capped to 100/number of CPUs on multi CPU Windows machines. This has now been fixed.
CR213687, CR213685	The non-supported option <code>-XXprintStackOverflow</code> has been added. This option produces a full stackdump when the <code>StackOverflowError</code> is thrown.
CR218035, CR230226	In the previous release, BEA JRockit was known to hang or crash on 2.6 kernels on Itanium, due to a bug in the Linux 2.6.11 (and previous) kernel. The bug has now been fixed in kernel 2.6.12 by the Linux vendors. Note: To run this release of BEA JRockit, you need to use SLES 9 SP2 or RHEL4 U1 (or later).
CR219610	The default freelist cache size is 10% of the current heap size (with a minimum of 3 MB).
CR225145	Changes to <code>java.vendor*</code> system properties. The correct values are: <ul style="list-style-type: none"> <code>java.vendor</code> = "BEA Systems, Inc." <code>java.vendor.url</code> = "http://www.bea.com/" <code>java.vendor.url.bug</code> = "http://support.bea.com"
CR226460	The experimental code cache feature has been removed due to stability issues.
CR228592	Using the Memory Leak Tool could in some instances make JRockit freeze or crash. This has now been fixed.
CR228822	When running BEA JRockit on a single CPU machine, the code optimizer was in some cases too intrusive (true for BEA JRockit 5.0 R25.2.0). The problem was sometimes noticed at the first start of the WebLogic console. This problem has now been fixed. In the previous release of BEA JRockit, this problem was worked around by setting the flag <code>-Djrockit.codegen.optpriority=1</code> ; if you are using this flag, remove it when updating to this release.
CR229981	Improved behavior of internal locks that can lead to better performance during heavy loads.

Change Request ID	Description
CR230236	The <code>-Xmanagement</code> option resulted in an overhead even though the BEA JRockit Management Console was not connected. This has now been fixed.
CR232847	Improved floating point performance.
CR235100	<p>The <code>java.vm.version</code> for the previous release was: dra-45238-20050523-2021-win-ia32</p> <p>The <code>java.vm.version</code> for this release is: R26.0.0-188-52875-1.5.0_04-20051110-0917-win-ia32</p> <p>The <code>java.vm.info</code> for the previous release was: R25.2.0-28</p> <p>The <code>java.vm.info</code> for this release is: <empty></p>
CR235101	Previously, BEA JRockit calculated <code>MemoryMXBean.getNonHeapMemoryUsage()</code> used as the process virtual bytes minus the heap size. Now <code>MemoryMXBean.getNonHeapMemoryUsage()</code> used is calculated as the process rw memory minus the heap size.
CR235105	The heap occupancy trigger heuristics have been corrected.
CR235107, CR236922	<p>When running JRockit with a concurrent garbage collector, the garbage collector starts before the heap is completely full to be able to finish the garbage collection before running out of heap memory.</p> <p>tries to determine when a garbage collection needs to be triggered through heuristics, but in certain situations it might be beneficial to set this trigger by hand and to a fixed value. Use the following argument <code>-XXgctrigger=<int></code>, where <code>int</code> is an integer that takes values between 0 and 100. The value specifies the amount of free heap, measured in percentage, that should be available for the argument to trigger.</p>
CR235682	Previously selecting a generational, concurrent mark, concurrent sweep strategy resulted in a non generational, parallel mark, parallel sweep strategy being chosen. This has now been fixed.
CR236723	A warning appears at start-up if you are running with a “suspicious” thread system.

Change Request ID	Description
CR237093	<p>The time for the reference update phase was measured incorrectly when running the parallel garbage collector. This made the statistics that the (compaction) heuristics are based on incorrect.</p> <p>The pause target for compaction increased too fast when running static garbage collections. In this release, it can be increased with at most 50% for each garbage collection. The initial value has been set to 100 ms (this may be further tuned).</p>
CR238220	CPU load and CPU description is now returned as CompositeData in JRockitConsoleMXBean.
CR239499	The Memory Leak Detector did not work properly when the management server was started by the Ctrl-Break handler (using <code>ctrlhandler.act</code>) as opposed to using the <code>-Xmanagement</code> startup option. This has now been fixed and the Memory Leak Detector works as expected, regardless of how the management server has been started.
CR239968	<p>In previous versions of , the maximum stack trace depth value, in JRA recordings, was always 16 frames. In this version it is possible to set this value by adding the “tracedepth” option to <code>-XXjra</code> and the “jrarecording” <code>ctrlbreak</code> handler.</p> <p>The default value is still 16 frames.</p>
CR240355	In this release it is possible to change verbosity for the “memory,” “memdbg,” and “gcpause” subsystems using the <code>ctrlbreak</code> handler.
CR240359	<p>The syntax for the “verbosity” <code>ctrlbreak</code> handler has changed. The previous argument “args” has been changed to “set.”</p> <p>Run the <code>ctrlbreak</code> handler “help verbosity” for more details.</p>
CR240510	<ul style="list-style-type: none"> The information from <code>jrockit.verboserefs</code> has been improved and now includes more details regarding garbage collection. Support for verbose information in <code>ctrlbreakhandler/jrcmd</code> has been added.
CR241377	The default stack size on Solaris/Sparc is 256k.

Change Request ID	Description
CR241546	<p>This release of BEA JRockit does not ship with Java Web Start or Java Plugin. Some earlier releases did and the installers and uninstallers of those versions do not behave properly when this release is installed. To avoid problems with the installation, do one of the following before installing :</p> <ol style="list-style-type: none"> 1. Uninstall all earlier BEA JRockit JRE releases before installing this release. Java Web Start and Java Plugin will not be available after this process. 2. Install all earlier BEA JRockit JRE releases that are needed before installing this release. Java Web Start and Java Plugin will be available if included in any of the earlier releases. <p>Note: Do not install or uninstall an earlier release of BEA JRockit JRE while this release is installed. Doing so may corrupt the state of Java Web Start and Java Plugin.</p>
CR241638	<p>The following compaction tuning flags have been added:</p> <p><code>-XXinternalCompactRatio</code> Sets the number of heap parts to compact during internal compaction. Default is dynamic or 8 when running <code>-Xgcprio:throughput</code>.</p> <p><code>-XXexternalCompactRatio</code> Sets the number of heap parts to compact during external compaction (aka "evacuation"). Default is dynamic or 8 when running <code>-Xgcprio:throughput</code>.</p> <p><code>-XXheapParts</code> Sets the number of heap parts. Default is 128.</p> <p>Furthermore, the system property <code>jrockit.gc.usematrix</code> has been turned into an <code>-XX</code> option.</p> <p><code>-XXusePointerMatrix</code> Indicates that the pointer matrix should be used instead of the pointerset. The pointer matrix is default when running <code>-Xgcprio:deterministic</code> or <code>-Xgcprio:pausetime</code>.</p>
CR241665	<p>In the management API, the functions <code>getMAC</code> and <code>getMTU</code> are supported on Windows. On Unix systems these functions return an empty string or zero.</p>
CR242307	<p>To get fixes for potential security vulnerabilities, this release has upgraded zlib from <code>zlib-1.2.1</code> to <code>zlib-1.2.3</code>.</p>
CR242944	<p>The command <code>jrockit.oomdiagnostics.filename</code> specifies where to write out of memory diagnostics (if this is enabled through <code>jrockit.oomdiagnostics</code>). If diagnostics are enabled and no file is specified, the output ends up where the <code>-Xverbose</code> information ends up (typically <code>stderr</code>).</p>

Change Request ID	Description
CR244403, CR238634	Traversing superclasses of interfaces in <code>find_method</code> have been removed. These classes returned methods that were not declared in the interface or its super interfaces, for example, <code>Object.*</code> . The supermost class of an interface is always <code>Object</code> .
CR245707	JVMDI is not supported in the BEA JRockit R26.0 release (nor in the 25.0 or 1.4.2 builds); however, JDWP and JDI are supported. This means that remote debugging tools will still work as in previous releases.
CR245732	Previously, retrieving JMAPI stack traces could deadlock in certain cases for traces that included overridden <code>hashCode</code> methods that had been taking locks. This has now been fixed.
CR255294	Previously, when calling <code>java.io.File.getCanonicalFile()</code> on a path where <code>java.io.FilePermission</code> was not granted, the call did not fail as expected. This has now been fixed and the appropriate exception is thrown.

Known Issues

The following issues are known in BEA JRockit R26:

Issue	Description
BEA JRockit is crashing due to a signal handling conflict. <i>For Linux users only.</i>	<p>If you are using BEA JRockit in conjunction with a native library that relies on OS signals you may experience crashes due to a signal handling conflict between BEA JRockit and the native library.</p> <p>Workaround:</p> <p>Set the environment variable <code>LD_PRELOAD</code> as follows:</p> <pre>export LD_PRELOAD=\$JROCKIT_HOME/jre/lib/i386/libjsig.so</pre> <p>BEA Engineering found this conflict using IBM's MQSeries native drivers, and it may be present in other libraries that rely on native code.</p> <p>For more information, see:</p> <p>http://java.sun.com/j2se/1.5.0/docs/guide/vm/signal-chaining.html</p>
CR289856	<p>Due to a bug in the Jikes 1.22 Java compiler, it cannot be used for compiling 1.4.2 Java code using the class libraries shipped with BEA JRockit R26.0.0 or newer releases.</p> <p>The bug has been reported and is described at:</p> <p>http://sourceforge.net/tracker/index.php?func=detail&aid=1548770&group_id=128803&atid=712760</p>

Issue	Description
CR286338	In BEA JRockit 26.4, JRA Recordings always report that the heap usage is zero after a garbage collection with a single-spaced garbage collector. The <code>-Xverbose:memory</code> and <code>-Xverbose:memdbg</code> printouts report the correct value for the heap usage after garbage collection.
CR284602	The <code>jrcmd</code> tool might fail to find all BEA JRockit processes on a machine if there are any Java processes by other JVM vendors running on the same machine. Contact BEA JRockit Support for a patch that solves this problem.
CR283915	<p>Long thread sleeps issued by <code>Thread.sleep(...)</code> and <code>Object.wait(...)</code> can end too early if the sleeps are longer than <code>0x3FFFFFFF</code> milliseconds (approximately 12.4 days).</p> <p>All platforms are affected.</p>
CR283787	<p>Upgrading from BEA WebLogic Platform 8.1 SP5 to 8.1 SP6 and running it on BEA JRockit R26 SP3 can result in a performance regression of up to 10%.</p> <p>Workaround:</p> <p>To avoid this regression, you can improve the performance of memory intensive applications by setting the command-line options <code>-XXtlaSize:<default 2kB></code> and <code>-XXlargeObjectLimit:<default 2kB></code>. See the “Special Note for WLP 8.1 SP6 Users Running on BEA BEA JRockit R26.3” for instructions on how to do this.</p>
CR280443	<p>When you expand types in the Type Graph in the BEA JRockit Memory Leak Tool, BEA JRockit can hang and become unresponsive while consuming CPU resources. This issue was introduced as a regression in BEA JRockit R26.2.</p> <p>Note: This has been fixed in R26.4.</p>
CR279998	Objects that are allocated with reflection, for example, with <code>java.lang.Class.newInstance()</code> , do not show up in the allocation stacktraces in the Memory Leak Detector Tool.
CR279584	<p>The synchronization code in <code>java.util.Random.next()</code> has been optimized for the case where there is no contention, i.e. the <code>java.util.Random</code> object is used by only one thread. A drawback of this optimization is worse performance if the object is used heavily in several concurrent threads. This typically happens if the convenience method <code>java.lang.Math.random()</code> is used.</p> <p>To avoid this, create a new <code>java.util.Random</code> object instead of calling <code>java.lang.Math.random()</code>.</p>

Issue	Description
CR278796	<p data-bbox="475 541 1187 567">Null-pointer exception bypasses first catch block and is caught in the next.</p> <p data-bbox="475 583 1304 663">A catch block immediately following “return x.y;” where the variable x points to null, will not catch the null-pointer exception. The exception will be caught in the next catch block.</p> <pre data-bbox="475 684 1065 1329"> public class Test { int y = 0; public static int foo() { try { Test x = null; return x.y; } catch (Exception e) { System.out.println("It works!"); } return 0; } public static void main(String[] args) { try { foo(); } catch (Exception ex) { System.out.println("Failure!"); } } } </pre>
CR276311	<p data-bbox="475 1354 1304 1409">BEA JRockit R26.3.0 can, under rare circumstances, crash when using -Xgc:gencon. All platforms are affected.</p> <p data-bbox="475 1423 618 1449">Workaround:</p> <p data-bbox="475 1463 1114 1488">Use another garbage collector or upgrade to BEA JRockit R26.4.0.</p>
CR275524	<p data-bbox="475 1516 1284 1596">On some Linux versions, the library functions exit () and _exit () do not always terminate the calling process. This causes BEA JRockit to hang during shut down on SLES 8.</p> <p data-bbox="475 1612 618 1638">Workaround:</p> <p data-bbox="475 1652 721 1677">Use a later SLES version.</p>

Issue	Description
CR274636	<p>BEA JRockit 1.4.2 R26.2 and R26.3 throw a <code>java.lang.NullPointerException</code> when passing a null argument to <code>java.lang.String.getBytes(String charsetName)</code>, while BEA JRockit 1.4.2 R24 did not. The behavior is unspecified but the change might cause problems when running TIBCO with XML messaging on BEA JRockit 1.4.2 R26.2 or R26.3.</p> <p>Note: This has been fixed in R26.4.</p>
CR272699	<p>Occasionally, when using a JVMTI Java debugger, breakpoints are not hit. This issue can arise with any R26 version of the product.</p> <p>Workaround:</p> <p>Start BEA JRockit with the option <code>-XXnoCodeGC</code>.</p> <p>Note: <code>-XXnoCodeGC</code> is only intended for troubleshooting and is not recommended nor supported for production use.</p>
CR271551	<p>Buffers that have been allocated through <code>ByteBuffer.allocateDirect()</code> would not be released, even when discarded by the application, which causes C heap memory leaks.</p> <p>Note: This has been fixed in R26.4.</p>
CR269115	<p>BEA JRockit crashes when optimizing method. This issue can be identified by the following stack trace:</p> <pre>at renameVar+36()@... at irCompactVars+240()@... at ssaConvertTo+2356()@... ...</pre> <p>This crash has only been noted on the Solaris on Sparc platform but might be a problem on other platforms as well.</p> <p>Workaround:</p> <p>Use <code>optfile</code> and remove the particular method that causes the crash.</p>
CR268746	<p>On older Linux distributions that run <code>LinuxThreads</code> instead of <code>NPTL</code>, BEA JRockit can sometimes hang when it is shutting down.</p>
CR268439	<p>Calling JVMTI functions from a non-attached thread will cause BEA JRockit to crash.</p> <p>Note: This has been fixed in R26.4.</p>
CR268423	<p>BEA JRockit releases previous to R26.3.0 <i>do not</i> contain the fix for the Australian Daylight Savings Time change for 2006. Please contact BEA Support for a patch</p>

Issue	Description
CR267987	<p>Using <code>java.nio.channels.SelectionKey.OP_CONNECT</code> will make BEA JRockit block forever.</p> <p>Note: This has been fixed in BEA JRockit R26.4.</p>
CR266871	<p>BEA JRockit R26.0.0 on Linux IA32 can experience problems setting up memory for object allocation. It will manifest itself through this printout (and then exit BEA JRockit):</p> <pre>[JRockit] ERROR: Fatal error in JRockit during memory setup phase. Try to reduce the heap size using -Xmx:<size>m, i.e. "-Xmx:16m". Could not create the Java virtual machine.</pre> <p>Workaround:</p> <p>Try different <code>-Xmx</code> values to find a heap size that is setup correct.</p> <p>Note: This known issue is valid for R26.0.0. The problem is fixed in releases R26.1.0 and later.</p>
CR266870	<p>IA64 RedFlag 4.1 creates broken core files when programs crash. This makes it impossible for BEA JRockit engineers to resolve customer issues on RF41/IA64.</p> <p>This has been reported at: http://www.redflag-linux.com/bugzillaAX/show_bug.cgi?id=1320</p>
CR266667	<p>Slow startup because of a hang in <code>java.net.PlainSocketImpl.initProto()</code>, which typically is called when creating the first <code>Socket</code> or <code>ServerSocket</code>.</p> <p>In BEA JRockit 5.0 R26 the network stack is configured so that IPv6 is used in preference to IPv4 when it is present.</p> <p>During initialization of the network stack, the network code connects a socket to its own loopback interface to set up some data structures. Blocking this connection (e.g. with a firewall) will cause the initialization code to wait for a socket timeout, after which the system falls back on using IPv4.</p> <p>Workaround:</p> <p>Either set <code>-Djava.net.preferIPv4Stack=true</code>, which forces Java to use IPv4 instead, or you disable IPv6 entirely in the system. The proper fix is to allow IPv6 traffic from localhost to localhost.</p> <p>For more information, see the Sun documentation: http://java.sun.com/j2se/1.4.2/docs/guide/net/ipv6_guide/#ipv6-networking</p>

Issue	Description
CR265793	<p>With JRockit 1.4.2 R26, <code>java.lang.reflect.Array.set(Object array, int index, Object value)</code> always throws <code>NullPointerException</code> when value is null without checking if array is of primitive type.</p> <p>A patch addressing this issue is available through BEA support.</p>
CR265227	<p>The BEA JRockit in Silent mode will not be correctly installed if the <code>USER_INSTALL_DIR</code> in the XML-file has been set to anything other than the default installation path.</p> <p>During the installation, the registry settings will not be set correctly, causing the <code>.jar</code> file association to fail. The files <code>Java.exe</code> and <code>Javaw.exe</code> will not be copied to <code>%SystemRoot%\System32</code> either.</p> <p>Note: This problem has been fixed in R26.3.</p>
CR264913, CR244553	<p>A bug in the Linux operating system on x64 will cause BEA JRockit to crash if it is invoked from the <code>pthread_once</code> system call.</p> <p>Workaround:</p> <p>Install RHEL 4.0 QU3 or SUSE 9.0 SP3.</p>
CR262540	<p>An issue running HP OpenView Java Diagnostics Profiling Agent may cause crashes with BEA JRockit R26.0.0.</p> <p>Note: This issue has been fixed in R26.3.0.</p>
CR262157	<p>In rare cases BEA JRockit can print incorrect information about locks in the stack dump, given by the Ctrl-Break handler or by the Management Console/MAPI.</p> <p>When a lock is taken by a method that has been optimized in a certain way (inlining), this lock can be printed as being taken, not only on the correct frame, but also as being taken on one or several nearby frames. This does not affect how BEA JRockit treats the lock when executing, only the stack dump itself.</p>

Issue	Description
CR256312	<p>On Windows x64 and Itanium, when using the BEA JRockit JRE console mode installer to remove a previously installed BEA JRockit, the uninstall will be interrupted and the following message is displayed:</p> <p>A fatal error has occurred. This application will terminate.</p> <p>The uninstall information is now removed, but most of the files and registry settings are still left on the machine.</p> <p>Workaround to completely uninstall the BEA JRockit JRE:</p> <p>Run the installer in graphical mode and click Remove previous and reinstall when prompted; the BEA JRockit JRE is once again installed. To completely uninstall the BEA JRockit JRE (including its files and registry settings), use the graphical uninstall procedure.</p>
CR252610	<p>A problem regarding the handling of TrueType fonts for certain font files may cause a call to <code>java.awt.Font.getXXX()</code> methods that results in an <code>IllegalArgumentException</code> being thrown.</p> <p>This problem has been reported to Sun as a problem found in Sun JDK 5.0 Update 4 in bug #6349101 (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6349101). This problem can theoretically occur on all platforms but has been observed on RedFlag Linux Advanced Server 5.0 when the RPM package <code>ttfonts-zh_TW-5.0-2AX.noarch.rpm</code> is installed and the user is requesting a font with Chinese locale.</p> <p>Workaround:</p> <p>Uninstall the package <code>ttfonts-zh_TW-5.0-2AX</code>. This solves the problem but will at the same time remove those fonts from the system, which may cause other problems when trying to display Chinese text. If you try this, you should try to replace the use of the removed font with some other available font on the system.</p> <p>Note: The problem has been fixed in release R26.3.0, but since that release does not include an Itanium version, it is still a known issue for the R26.0.0 Itanium version.</p>
CR251457	<p>If <code>ulimit -v</code> is used on Linux to limit the virtual memory usage, BEA JRockit may crash if the limit is set too low. The x64 version of BEA JRockit requires a much larger setting because it reserves addresses for compiled code at startup. The BEA recommendation is to not use the <code>ulimit -v</code> setting at all.</p>

Issue	Description
CR251452	<p>BEA JRockit needs enough virtual memory (address space) to be able to run properly. Note that a high value on allowed virtual memory does not imply high memory consumption. On Linux, the amount of available virtual memory can be changed, typically by the command <code>ulimit -v <amount></code>.</p> <p>The default is an unlimited address space, which is the recommended. Do not change this default, since it risks having BEA JRockit running out of address space, which in turn causes BEA JRockit to terminate immediately.</p> <p>BEA JRockit requires at least the following amount of virtual memory to even start:</p> <ul style="list-style-type: none"> • On IA64: 88 MB • On IA32: 61 MB • On x64: 1100 MB <p>Note: If you start BEA JRockit with less virtual memory than twice the above values, you'll be given a warning. This is not recommended, and is likely to cause problems when running an actual Java application.</p>
CR249667	<p>Some applications may experience problems with the automatic nursery sizing heuristics when running in <code>-Xgcprio:throughput</code> (default) and <code>-Xgcprio:pausetime</code> mode. This causes too frequently triggered garbage collections.</p> <p>Workaround:</p> <p>Set the nursery size manually using <code>-Xns:<size></code> or to select a static garbage collector.</p> <p>The automatic heap resizing heuristics are not optimal for all applications. If your application has problems due to frequent garbage collections and the heap hasn't been expanded to the maximum heap size, you can increase the initial heap size (<code>-Xms:<size></code>) to improve the performance.</p>
CR248565	<p>When running BEA JRockit as an embedded service on Windows (for example, Jakarta Tomcat service wrapper), the directory <code><path-to-jdk>/jre/bin</code> must be added to the PATH variable.</p> <p>For more information, see:</p> <p>http://edocs.bea.com/jrockit/geninfo/devapps/tshoot.html#998592</p>
CR248551	<p>When installing the 32-bit JRE on a Windows x64, the <code>java.exe</code> and <code>javaw.exe</code> files will not be copied to the system32 folder; they will be placed in the syswow64 folder instead.</p> <p>This is expected behavior for 32-bit applications running on Windows x64.</p>

Issue	Description
CR247613	The “jrcmd” utility shipped in the Windows ia32 BEA JRockit package does not work on Windows x64. Instead, use the “jrcmd” utility shipped with the Windows x64 package of BEA JRockit.
CR246634	Thread priorities are supported on the Windows platforms only.
CR246224, CR260004	<p>When doing a JRA recording on Windows XP x64, the JRA recording incorrectly displays that it has been done on Windows 2003 Server. The current implementation of BEA JRockit cannot separate between the two different operating system’s version information.</p> <p>Note: This known issue is valid for R26.0 and R26.1.</p>
CR245914	<p>Java applications (such as Eclipse) may hang on Linux distributions which use the “gamin” File Alteration Monitor implementation, for example RHEL4. This is due to a bug in gamin’s handling of signals.</p> <p>Workaround:</p> <p>Use “signal chaining” by loading the libjsig.so library. Do this by executing</p> <pre>> export LD_PRELOAD=\$JDK_HOME/jre/lib/i386/libjsig.so</pre> <p>before starting BEA JRockit.</p> <p>For more information on signal chaining see: http://java.sun.com/j2se/1.5.0/docs/guide/vm/signal-chaining.html</p>
CR244773 CR250025	<p>BEA JRockit’s nursery pool can be invalidated when the garbage collection strategy changes. According to the Java 2 Platform SE API documentation, a pool that has been invalidated may return null. Programmers must therefore assume that</p> <pre>MemoryPoolMXBean#getUsage()</pre> <p>and</p> <pre>MemoryPoolMXBean#getPeakUsage()</pre> <p>may return null at any time.</p> <p>The MemoryMonitor demo and the VerboseGC demo can throw <code>NullPointerException</code>s, since these attributes are not checked for nulls.</p>
CR243996	<p>The VerboseGC demo, located at</p> <pre>\demo\management\VerboseGC\VerboseGC.jar</pre> <p>, can throw a <code>NullPointerException</code> when used with a BEA JRockit that has a nursery.</p> <p>According to the API specification for the Java 2 Platform Standard Edition 5.0 the method <code>getUsage()</code> of the <code>MemoryPoolMXBean</code> can return null if a memory pool is not valid, which can be the case when you run BEA JRockit with a nursery.</p> <p>This validity check is missing in the demo and is the cause of the <code>NullPointerException</code>.</p>

Issue	Description
CR242655	<p>In Windows, faulting code can be caught by Structured Exception Handling (SEH). The Microsoft C compiler allows a special construct, see below:</p> <pre>__try { // do something that can fail } __except (filterException()) { // handle the fault }</pre> <p>This sets up an SEH handler, which would get called if the code in the <code>__try</code> block fails (for example, a read/write to an illegal address).</p> <p>However, on 64-bit Windows (IA64 and x64), BEA JRockit uses a new exception handling feature known as vectored exception handlers. The vectored exception handlers will be called before any SEH handler gets called. If the BEA JRockit vectored exception handler detects a fault in native code, it will make BEA JRockit produce a crash dump.</p> <p>The effect of this is that you cannot use SEH on 64-bit Windows in native code that gets called by BEA JRockit. Either install a vectored exception handler yourself and add it first in the chain, or test the memory before trying to read/write to it with <code>IsBadReadPtr()</code>.</p>
CR232872	<p>On RHEL3u6 and earlier, as well as RHEL4u2 and earlier, <code>fork()</code>ing new processes can sometimes fail. This can in turn make <code>Runtime.exec()</code> fail. This has been fixed in RHEL3u7 and RHEL4u3.</p> <p>The Redhat Issue Tracker case number for this is 77560. This isn't available in Redhat's public bugzilla.</p>
CR210743	<p>If you are running SLES 8.0, RFAS 4.1, RHEL 3.0 QU4 or older, you might run into serious IO problems.</p> <p>Workaround (for RHEL):</p> <p>Install version QU5 or later.</p>
CR128962	IPv6 support for Windows is included as an unsupported feature in this release.

