# BEA JRockit JDK

## Release Notes

JRockit 5.0 Service Pack 2
June 2005

# Copyright

# Restricted Rights Legend

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

# Contents

# BEA JRockit 5.0 JDK Release Notes

**Version 5.0 (R25.2) JDK**

This document contains important details for BEA JRockit 5.0 (R25.2) JDK. It contains information on the following subjects:

## License Agreement

The BEA JRockit 5.0 (R25.2) JDK is subject to the terms and conditions of the BEA JRockit Binary License Agreement.

# Platform Support

- The BEA JRockit 5.0 (R25) JDK release is certified on the platforms listed on the BEA JRockit Supported Configurations page.

# Java Support

- The BEA JRockit 5.0 (R25.2) JDK release is certified to be compatible with J2SE5.0 Update 3.
- The BEA JRockit 5.0 (R25.1) JDK release is certified to be compatible with J2SE5.0 Update 2.
- The BEA JRockit 5.0 (R25.0) JDK release is certified to be compatible with J2SE5.0.

# Installation

BEA JRockit JDK is available as a standalone application. For instructions on installing BEA JRockit, please see Installing BEA JRockit JDK.

# Other Documentation

This release of BEA JRockit JDK includes a complete documentation set comprised of these documents:

- BEA JRockit JDK Release Notes
- Introduction to BEA JRockit
- Using BEA JRockit JDK
- Using BEA JRockit Runtime Analyzer
- Using BEA JRockit Management Console
- Tuning BEA JRockit JVM
- Developing Java Applications
- Supported Platforms
- Installing BEA JRockit JDK
- Developers FAQ

Copies of all BEA JRockit documents can be found at:

http://e-docs.bea.com/wljrockit/docs50/index.html

# Most Recent Changes

Below you find information about the most recent changes of BEA JRockit.

## Changes in JRockit 5.0 (R25.2)

The following CRs have been corrected for this release.

| Change Request ID | Description |
|---|---|
| CR233415 | Looking at allocation stacktraces in the Memory Leak Detector could show stacktraces where the selected type was not allocated, as well as stacktraces where it was allocated. This has been fixed. |
| CR224247 | An issue that prevented the Java Plugin from running applets in Internet Explorer is solved in this version. JRockit 5.0 will now run applets in Internet Explorer. |
| CR214904 | Under high network load, JRockit could sometimes incorrectly throw exceptions of the type “java.net.SocketException: Socket closed.” This has now been fixed. |
| CR224896 | Options that are not valid JRockit options, have sometimes been parsed as a different, valid option. This happened if the valid option began with the same characters as the invalid one. For example, `-Xcheck` could be parsed as `-XcheckedStacks`. This problem is now fixed. |
| CR204924 | `FileChannel.read` now returns -1 at end of file instead of throwing an exception. |
| CR224377 | Calling Object.wait() in a system with high lock contention could cause JRockit to throw a NullPointException. This has now been fixed. |
| CR225425 | The Memory Leak Detector now lists the top 10 items (if applicable) when listing the largest arrays. |
| CR226240 | The JRockit Analyzer (JRA) does not crash the JVM anymore when the JVM runs with the gencon garbage collector. |
| CR227186 | The garbage collector MXBean method `getNurserySize` does not throw an internal RuntimeException anymore when a nursery is not available. |
| CR206727 | JRockit will no longer crash when installed as a web browser plugin and you press “v” in the Java Console. |

| Change Request ID | Description |
|---|---|
| CR215677 | Hour-glass does not stay on the screen in Management Console anymore. |
| CR221097 | JRockit issues events on multiple threads during garbage collection. New option, `-Xjvmpi:singlegc=true`, has been added to force events being issued via one thread only. |
| CR221238 | JRockit memory usage has been improved in cases where there are many classloaders with few classes per classloader. |
| CR221635 | User defined actions in the Management Console are now working. |
| CR221996 | JRockit startup time has been improved. |
| CR222428 | Stack overflow issues are now handled more safely. |
| CR222443 | Previously JRockit displayed incorrect nursery size information when using `-Xverbose:memory` with an optimizing garbage collector. This information is no longer displayed. |
| CR222539 | The optimizing compiler could generate inaccurate code for `StringReader.read`. This problem is now fixed. |
| CR222737 | A race condition that could lead to a null pointer exception when generating the detail panes in the Management Console is now eliminated. |
| CR222973 | Prevent applications from changing the private field parent in `java.lang.ClassLoader`. |
| CR222990 | Can now create arrays larger than 256*1024*1024 elements. |
| CR223407 | The `MemoryMBean` attribute `ObjectPendingFinalizationCount` now returns the correct value. |
| CR223426 | On 64-bit platforms, JRockit could crash if the Memory Leak Detector tried to retrieve the largest arrays for a type. This has now been fixed. |
| CR223640 | When using `IterateOverReachableObjects` in JVMTI, JRockit reported bad referrer_index for JVMTI_REFERENCE_FIELD. This has now been fixed. |
| CR225144 | Reverted the `java.vendor*` system properties back to BEA specific property values. |
| CR225152 | Eliminated sporadic deadlocks on platforms with weak memory guarantees. |
| CR226873 | Previously JRockit could crash during shutdown, possibly without generating either a crash dump or a core file. This was an uncommon problem, but has now been fixed. |

## Changes in JRockit 5.0 (R25.1)

The following CRs have been corrected for this release.

| Change Request ID | Description |
|---|---|
| CR209583 | CGLIB does not throw `IllegalAccessError`. |
| CR211573 | JRockit on x64 may crash due to a floating point code generation bug. This is now fixed. |
| CR211938 | Overriding package local final Methods now works. |
| CR211961 | `java.vm.info` property now exists. |
| CR212193 | When iterating over stack frames, `getStackAccessControlContext` used the wrong iterator (frame instead of call), which could result in JRockit missing optimized frames. |
| CR212198 | Throwing `ZipException` instead of `IOException` when encountering empty or invalid zipfile |
| CR212423 | System.arraycopy had an off-by-one error with short-arrays when doing a reverse copy. This is now fixed. |
| CR214656 | `lphello` does not dump in `cgGetCallChainOnAddr`. |
| CR215089 | Large page for IA32 Linux with 2.6 kernels are now working. |
| CR215143 | Problems with method resolving in JNI now fixed. |
| CR215444 | WebLogic Enterprise Platform `arraycopy` might read outside of heap on X86. This issue is now fixed. |
| CR217645 | JRockit is now able to throw `ClassCircularityError`. |
| CR218157 | Optimizing problems causing a thread dump, sometimes occurring with `optimizeIt`, now fixed. |
| CR219032 | Heap/object allocation scaling bottleneck solved by modifying `-XXaggressive` to set the cachelistpercentage to 10% and by adding `jrockit.heap.cachelistpercentage` property. This sets the percentage of free heap memory that will be kept in the freelist caches that are used for speeding up object allocation (for example set: `-Djrockit.heap.cachelistpercentage=15` on the command line to set the percentage to 15%). |

| Change Request ID | Description |
|---|---|
| CR220374 | JRockit dump in `zipReleaseEntry` during `adminServer` (WLS) startup, R25.1.0-96. This is a known issue in a third-party product used by BEA JRockit. |
| CR220655 | Rare “timeout” with `gencon` showed out to be an internal signal error. This is now fixed. |
| CR220819 | In the BEA Installer the Java Control Panel is now installed. |
| CR220825 | If JRockit is uninstalled and thereafter reinstalled, the installer fails to set JRockit as a plugin to any browser. This problem has now been fixed. |
| CR221005 | BEA Installer: Multiple instances of JRockit installation problem has been fixed. |
| CR221009 | If user installs JRockit in another location than the default location, the Registry keys still point to the default location (which do not exist). This problem is now fixed. |
| CR221016 | JRockit does not fail to install Java Web Start anymore. |
| CR221029 | Java Application Cache Viewer can now be launched from `bin` as well as `jre/bin`. |
| CR222056 | Internet Explorer is now shown as a browser alternative. |
| CR194599 | Support for profiling of VM internal locks are now implemented. |
| CR195900 | External system properties handling is now cleaned up. |
| CR205334 | `sun.reflect.ReflectionFactory.newConstructorForSerialization` does not work.<br>**Note:** JRockit does not support that API. |
| CR205501 | IPF largepages `MAP_PRIVATE` is now supported. |
| CR206811 | Invocation compiler `newInstance` now generates call to `registerfinalizer`. |
| CR206862 | Performance degradation over time with `-Xgcprio:throughput` due to incorrect strategy choice. Heuristics updated and no degradation can be observed. |
| CR207085 | JRockit crashed when starting a server with `OptimizeIt` due to differences between the handling of `startEvents` in JVMTI and JVMPI. This is now fixed. |
| CR207247 | Optimistically enable HT optimizations if processor supports it but OS cannot confirm.<br>To enable the optimistic approach, the `jrockit.cpu.ia32.ht` property has been added. |

| Change Request ID | Description |
|---|---|
| CR209071 | JRockit now returns the class with the most specific return type from `Class#getDeclaredMethod` when several matching methods are found. |
| CR209777 | JRockit VM seems to hang when attaching `optimizeIt` and trying to deploy a client. OptimizeIt uses `getthreadstate` frequently. The implementation of that method used to suspend native threads to retrieve information. Now JRockit does not need to suspend native threads when retrieving their status. |
| CR210093 | `ClassFormatError` with resin 3.0.10. Now allows multiple `LocalVar*Table:s` in any order. |
| CR210268 | `StackOverFlowError` while viewing JRA profile has been fixed. |
| CR211066 | Bytecode verification is now less strict to be compliant with the reference VM. |
| CR211177 | Potential deadlock in JVMTI/rawmonitors when using gcevents has been eliminated. |
| CR211205 | During shut-down, JRockit crashed when failing to acquire shared memory. This is now fixed. |
| CR211322 | `GetLineNumberTable` now returns `JVMTI_ERROR_NATIVE_METHOD` in case of native methods. |
| CR212065 | Upgrade to 1.5.0_02: Instances of `java.lang.Enum` are now never finalized, Sun bug 5098065. |
| CR212317 | Fixed race condition in `add_JVMPI_entryexit_hooks`. |
| CR212612 | JRockit can now bootstrap when launched from UNC path. |
| CR214838 | Fixed `ObjectMonitor` memory leak. |
| CR214982 | JRA now samples even if started with `-Xnoopt`. |
| CR215448 | Out of memory error even though heap could still be expanded. Now heap expansion has been corrected. |
| CR216068 | Missing JNI support, like JAWT, is now added. |
| CR216076 | File handle leak in `FileNativeIO.open` on directories is now fixed. |
| CR216086 | JRockit max heap size on win_ia32 can now increase beyond 1574304kb with /3GB parameter in boot entry. |
| CR216620 | Fixed rapid increase of heap usage after `IllegalMonitorStateException`. |

| Change Request ID | Description |
|---|---|
| CR216626 | Starting JRockit with the option `-Xcleartype` has no effect. The option is now deprecated and may be removed in a future release. |
| CR217056 | The `jvmtiGetTag` now works. |
| CR217060 | JVMTI now reports `JVMTI_HEAP_ROOT_THREAD` to the correct callback in `iterateover` objects. |
| CR217076 | Can now add static initializer to `Object`. |
| CR217100 | Remote JMX using authentication, file + password, now works. |
| CR217466 | Eclipse dumps when running on JRockit. This problem is now solved. |
| CR217647 | JRockit does not hang anymore when trying to throw `StackOverflowError`. |
| CR218061 | `jrcmd` now resolves path to JRockit executable correctly. |
| CR218251 | JRockit now supports the `-Xrs` option. |
| CR219600 | Large unused nursery and still getting an out of memory error. This problem has now been fixed. |
| CR199232 | Eternal garbage collection if an application stores too many finished threads. This has now been fixed and JRockit will not collect garbage eternally. |
| CR200091 | JGCW does no longer expose problems with finalizers. |
| CR206447 | The name of the `-XXhpc` option has changed to `-XXhpm` due to confusion with industry acronym for “high performance computing.” |
| CR208302 | No more `npe` when adding `StringBuffer.append(char[])` to method profiling. |
| CR208310 | Management Console does not deadlock anymore when reconnecting to a new VM. |
| CR208532 | Warn if the host computer swaps during collection. If `verbose:memory` is turned on, JRockit now warns if the total number of page faults during garbage collection is bigger than 5% of the number of pages in the heap. |
| CR209826 | `fix_javahome`, `realpath`, and `symlinks` cause `java.home` to be incorrect.<br>In this release, JRockit handles longer `realpaths` from `symlink` to `java.home`, which corrects this problem. |
| CR210867 | Previously JRockit missed to report some `compile_method_events`. This is no longer the case. |

| Change Request ID | Description |
|---|---|
| CR212649 | JRockit can only allocate 1500m heap on Windows 2003 IA32. The JVM performance counters are now initialized after the heap is allocated, which corrects this problem. |
| CR214354 | Occasional JRockit crash when starting a JRA recording has now been fixed. |
| CR214516 | The initialization behavior and value of sun.boot.class.path differs between JRockit and Sun. The initialization behavior has now been rewritten to mirror the behavior of Sun. |
| CR214537 | Getting threads by name from MAPI now works. |
| CR215113 | Cannot build update-site in Eclipse. |
| CR216646 | Counter for encountered deadlocks added. |
| CR217666 | The Management Console now shows deadlock detection data in thread stack dumps. |
| CR218492 | Parts of JVMTI system properties interface implementation now follow the specification. |
| CR208906 | A table is added for native lock profiling on the JRA's Lock Profiling page. |

# Features, Changes, and Enhancements

Below are new features, changes, and enhancements for each release listed.

## BEA JRockit 5.0 (R25.2)

- This release is mainly a stabilizing release.
- Tools documentation has gone through a major revision.

## BEA JRockit 5.0 (R25.1)

- Performance has been improved.
- Improved memory leak detection.
- Start-up time has been improved.
- The option `-Xrs` (reduce signal use) has been added. This option tells the VM not to use any operating system signals for a lot of things, but instead rely on the loading application (the launcher) to handle `TERM`, `INT`; `HUP`, and `QUIT`.

If you use the -Xrs option, you need to keep the following in mind:

a. JRockit will **not** run shutdown hooks on CTRL-C/SIGTERM, etc. (since it does not handle any of the signals).

b. On Linux, jrcmd will **not** work, since SIGQUIT is not handled.

c. The option CTRL-break/SIGQUIT will not work to get thread dumps.

**For Linux users**: Unless run with -Xrs the JVM will install handlers for SIGQUIT, SIGTERM, SIGINT and SIGHUP unless the loading process has set handlers for these to SIG_IGN, thereby hinting that these signals should not be used at all. If a calling process sets the handler for SIGQUIT to SIG_IGN, it will, however, still be replaced by the VM handler unless run with -Xrs.

## BEA JRockit 5.0 (R25.0)

- The dynamic garbage collector -Xgcprio:throughput is the default garbage collector.
- The startup flag -Xpausetarget has been added. This flag enables you to set a specific pausetime if you are using the dynamic garbage collector with pausetime argument. The default setting is 500ms.
- Support for J2SE 5.0, including support for the new JVMTI API, has been added. The JVMDI and JVMPI APIs are still supported.
- The BEA JRockit Management Console is now using the new management API in J2SE 5.0.
- Hardware Performance Counters and Dynamic Profile Guided Optimization (DPGO) have been added for Intel Itanium processors for improved performance. This is supported by SuSE Linux Enterprise Server 9.0 and later.
- The option -Xverbose can now be set with the arguments: opt, mem, gcpause, and gcreport.
- In previous releases of BEA JRockit, thread priority was default, but many customers ran into trouble with their applications; therefore, in this release, you need to set the thread priority flag if you want to use it. The flag is: -XXusethreadpriorities.

# Known Issues

The following issues are known in the BEA JRockit 5.0 release:

| Issue | Description |
|---|---|
| JRockit crashes when using a heap larger than 5375 MB on Red Hat 4.0 for EM64T/AMD64. | You can workaround this problem by using a heap smaller than 5375 MB.<br>This will be fixed in the next service pack (CR247853). |
| A signal being sent to wrong thread on Linux 2.6 kernels on Itanium (2.6.11 and previous Itanium kernels) might cause random hanging or crashing JVM. | BEA JRockit is known to hang or crash on 2.6 kernels on Itanium, due to a bug in the Linux 2.6.11 (and previous) kernels. The bug is in the kernel `sigprocmask()` syscall and will be fixed in the 2.6.12 kernel.<br>We cannot estimate how often the bug actually occurs, but under different circumstances it has been seen as often as once per hour or as seldom as once every two days.<br>BEA believes that this issue affects SMP systems more than single CPU systems.<br>Affected Linux versions are: SuSE Linux Enterprise Server 9.0 and RedHat Enterprise Linux 4.0 on Itanium running on 2.6.11 (or previous) kernels. (CR230226 and CR218035).<br>SuSE has confirmed that this bug will be fixed in their upcoming SLES 9.0 SP2 release. We are working with RedHat to get this fix into RHEL 4.0 Update 2.<br>You can see the committed patch at:<br>`http://kernel.org/git/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commit;h=a2a64769d0d3cc0380b4b6ecdcb781a2f790a69e`<br>Customers running SMP systems **must** patch their kernel to include the kernel fix or upgrade to a newer kernel. At the time of our release (June 22, 2005) no offically updated kernels are available. If you are running on SLES 9.0 Itanium, you should upgrade to SP2 once it is made available.<br>The RedHat internal reference number is 74397. For SuSE the internal reference number is 78084. |
| BEA JRockit Memory Leak Detector hangs when displaying aquired instances. | The BEA JRockit Memory Leak Detector could in some instances cause JRockit to freeze or crash. (CR228592) |

| Issue | Description |
|---|---|
| JRockit deadlock on SUSE Linux Enterprise Server 9.0 running on x64/SMP | There is a kernel bug in SUSE Linux Enterprise Server 9.0 x64 version, which causes signals to be lost on rare occations on multi-CPU systems. This is not noticable for most applications, but JRockit relies heavily on signalling, and is therefore affected by this bug. The consequence of the bug is that JRockit deadlocks. For this reason, it is *not recommended* to use JRockit for production on this platform, until this problem has been fixed by SUSE. BEA is working with SUSE to get the fix into the kernel.<br>Affected Linux version is SUSE Linux Enterprise Server 9.0 SP1. The problem is also present in RedHat Enterprise Linux 4.0 GA on x64, but it will be fixed in the upcoming RHEL 4.0 QU1. (CR230225 CR220658) |
| Crash in garbage collector under rare circumstances | JRockit may crash with an Illegal memory access and point to a problem in mmNurseryAddTLAsToCache. This is a timing dependent problem that might occur when run with a generational garbage collector. Low probability (CR229720). |
| Crashes in `cbGetCodeInfoOnAddr` | Under certain circumstances, JRockit can crash when unloading classes (CR229723). |
| JRockit may deadlock | If JRockit has been installed as a java plugin for a browser, pressing “v” (dump threads) in the Java Console has a high chance of causing JRockit to deadlock (CR225442) |
| `demo\jpda` trace example fails | This is because line numbers are not available for the classes in the `jrockit.jar` file (CR221291). |
| Corrupt or missing fonts or icons | When running `AWT/java2d/java3d` applications with JRockit, on certain configurations, the AWT DirectX acceleration might fail to initialize. This is due to virtual memory space conflicts.<br>**Workaround**: set an explicit `-mx` flag to approximately half the amount of the physical RAM or less. |

| Issue | Description |
|---|---|
| Static field values are reset to zero/null. | Using dynamic code replace from a debugger will cause static field values to be reset to zero/null, and can also cause spurious crashes at a later time due to un-updated references to the old field values.<br>**Workaround**: do not redefine class definitions containing static fields |
| JRockit is crashing due to a signal handling conflict.<br>For Linux users only. | If you are using JRockit in conjunction with a native library that relies on OS signals you may experience crashes due to a signal handling conflict between JRockit and the native library.<br>**Workaround**: set the environment variable `LD_PRELOAD` as follows:<br>`export LD_PRELOAD=$JROCKIT_HOME/jre/lib/i386/libjsig.so`<br>BEA Engineering found this conflict using IBM's MQSeries native drivers, and it may be present in other libraries that rely on native code.<br>For more information, see http://java.sun.com/j2se/1.5.0/docs/guide/vm/signal-chaining.html |