

Java SE Subscription Enterprise Performance Pack User's Guide



F55926-05



Java SE Subscription Enterprise Performance Pack User's Guide,

F55926-05

Copyright © 2022, 2024, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	iv
Documentation Accessibility	iv
Diversity and Inclusion	iv
Related Resources	iv
Conventions	v

1 Overview of Java SE Subscription Enterprise Performance Pack

Features and Enhancements of Enterprise Performance Pack	1-1
Installing Enterprise Performance Pack	1-2
Enterprise Performance Pack Version Strings	1-2
Configuring and Using Enterprise Performance Pack	1-3
Choosing a Garbage Collector	1-3
Running Tools and Using Libraries on Enterprise Performance Pack	1-4
Other Enterprise Performance Pack Differences	1-6

2 Changes to JVM Runtime Options

Options Added to Enterprise Performance Pack	2-1
Options Removed from Enterprise Performance Pack	2-2

3 Printing JVM Information

Preface

Java SE Subscription Enterprise Performance Pack is a runtime that delivers the performance of the JDK 17 Java Virtual Machine (JVM) to a Java SE 8 runtime.

Audience

This document is intended for users who want or need to continue using Java SE 8, are running Java SE 8 workloads at scale, and have an [Oracle Java SE Subscription](#) or are running on [Oracle Cloud Infrastructure](#).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- [Available Collectors](#) in *Java Platform, Standard Edition HotSpot Virtual Machine Garbage Collection Tuning Guide, Release 17*
- The [java command \(JDK 17\)](#) and the [java command \(JDK 8\)](#)

- [Oracle JDK 8 and JRE 8 Certified System Configurations](#)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview of Java SE Subscription Enterprise Performance Pack

Java SE Subscription Enterprise Performance Pack is a runtime that delivers the performance of the JDK 17 Java Virtual Machine (JVM) to a Java SE 8 runtime. For most situations, you can run Java SE 8 applications unchanged on Enterprise Performance Pack.

Enterprise Performance Pack improves performance and reduces the memory footprint for Java SE 8 workloads. It's ideal if you want or need to use Java SE 8, and you're running Java SE 8 workloads at scale. It makes better use of your processors and memory.

Enterprise Performance Pack is for server-side, headless systems (systems that operate without a graphical user interface or peripheral devices like a keyboard or a mouse) running 64-bit Linux (Intel or ARM); see [Oracle JDK 8 and JRE 8 Certified System Configurations](#) for more information. It's available as part of [Oracle Java SE Subscription](#) and [Oracle Cloud Infrastructure](#).



Note:

Enterprise Performance Pack is a runtime. It doesn't include JavaDoc API documentation. If you want to develop Java SE 8 applications, use Oracle Java SE Development Kit 8.

Topics

- [Features and Enhancements of Enterprise Performance Pack](#)
- [Installing Enterprise Performance Pack](#)
- [Configuring and Using Enterprise Performance Pack](#)

Features and Enhancements of Enterprise Performance Pack

Enterprise Performance Pack includes the following new features:

- [Z Garbage Collector](#): Is a scalable, low latency garbage collector.
- [Compact Strings](#): Is a space-efficient internal representation of strings, which reduces memory footprint and garbage collection activity; it's enabled by default.
- [Unified JVM Logging](#): Replaces JDK 8 options that print details about the JVM with `-Xlog` options. See [Printing JVM Information](#).

Enterprise Performance Pack includes the following enhancements:

- [Garbage-First \(G1\) Garbage Collector](#): Is the default garbage collector for Enterprise Performance Pack. It's targeted for multiprocessor machines scaling to a large amount of memory

- **G1 String Deduplication:** Reduces the memory footprint of `String` objects on the Java heap by taking advantage of the fact that many `String` objects are identical. It's disabled by default. Enable it with the `-XX:+UseStringDeduplication` option. See the `java` command in the JDK 17 documentation.
- **Class Data Sharing (CDS):** Helps reduce the startup time and memory footprint between multiple JVMs. In Enterprise Performance Pack, it's enabled by default. To disable it, see [Manually Controlling Class Data Sharing](#).
- **Flight Recorder:** Is a tool for collecting diagnostic and profiling data for a running Java application.

See [Running Tools and Using Libraries on Enterprise Performance Pack](#) for a list of tools and libraries that behave differently or are not supported on Enterprise Performance Pack. See [Other Enterprise Performance Pack Differences](#) for additional information.

Installing Enterprise Performance Pack

To install Enterprise Performance Pack, download and then unzip the `.tar.gz` Enterprise Performance Pack bundle.



Note:

Enterprise Performance Pack is available as part of [Oracle Java SE Subscription](#) Java SE subscription and [Oracle Cloud Infrastructure](#). It's available to customers for download through [My Oracle Support \(MOS\)](#) and on oracle.com/javadownloads.

The format of the Enterprise Performance Pack bundle name is `jdk-8u$UPDATE-perf-linux-x64.tar.gz`, where `$UPDATE` is the update version number. This number corresponds to the JDK 8 update version on which Enterprise Performance Pack is based.

Enterprise Performance Pack Version Strings

The Enterprise Performance Pack version strings contain the string `perf`. The command `java -version` prints these Enterprise Performance Pack version strings:

- The version of the Java SE 8 runtime on which it's based
- The version of the JDK 17 Java Virtual Machine on which it's based

The command `java -version` prints output similar to the following:

```
$ java -version
java version "1.8.0_345"
Java(TM) SE Runtime Environment (build 1.8.0_345-perf-4711-b06)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.4+1-perf-4711, mixed
mode)
```

Configuring and Using Enterprise Performance Pack

In most cases, you'll notice improved performance when you run your applications with Enterprise Performance Pack out-of-the-box. However, for optimal performance, configure Enterprise Performance Pack to use a suitable garbage collector. In addition, be aware that some tools and libraries behave differently compared to Java SE 8.



Note:

If you need additional help in using Enterprise Performance Pack such as assistance for a unique or exceptional use case or situation, contact [My Oracle Support](#).

Topics

- [Choosing a Garbage Collector](#)
- [Running Tools and Using Libraries on Enterprise Performance Pack](#)
- [Other Enterprise Performance Pack Differences](#)

Choosing a Garbage Collector

When first running a Java SE 8 application with Enterprise Performance Pack, try using the default garbage collector, Garbage-First (G1), with its default settings. In most cases, you should expect considerably better performance, even if you have fine-tuned your application with a JDK 8 garbage collector. Afterward, configure G1 as described in the JDK 17 documentation. Alternatively, consult the following table, which suggests a garbage collector to use depending on which one you're using in JDK 8.

Table 1-1 Suggested Garbage Collectors

Current JDK 8 Collector	Suggested Collector for Enterprise Performance Pack
Parallel Collector	Continue using the Parallel Collector by explicitly specify the <code>-XX:+UseParallelGC</code> option. Continue using the tuning options that you were using in JDK 8.
G1	The Garbage-First (G1) Garbage Collector is the default collector for Enterprise Performance Pack. Continue using it. You don't have to explicitly specify the <code>-XX:+UseG1GC</code> option.
Concurrent Mark Sweep (CMS) collector	CMS is not available in Enterprise Performance Pack. Use G1 with its default settings instead. Afterward, follow the tuning suggestions in Moving to G1 from Other Collectors . If you're unable to meet your application's latency or responsiveness requirements, then consider using Z Garbage Collector (ZGC) .
You're looking for a scalable low-latency collector	Use the Z Garbage Collector (ZGC) . Enable it with the <code>-XX:+UseZGC</code> option.

See [Selecting a Collector](#) in *Java Platform, Standard Edition HotSpot Virtual Machine Garbage Collection Tuning Guide, Release 17* for additional advice about selecting a garbage collector.



Note:

If you have any issues tuning your garbage collector for Enterprise Performance Pack, contact [My Oracle Support](#).

Running Tools and Using Libraries on Enterprise Performance Pack

Some tools and libraries behave differently on or have been removed from Enterprise Performance Pack. (Other tools and libraries that have been included in Enterprise Performance Pack should behave the same way as on JDK 8.)

The following JDK 8 tools behave differently on Enterprise Performance Pack

Table 1-2 Tools That Behave Differently on Enterprise Performance Pack

JDK 8 Tool	Behavior of Tool on Enterprise Performance Pack
<code>java</code>	Enterprise Performance Pack includes several runtime options from JDK 17. However, some options from JDK 8 are not available in Enterprise Performance Pack. See Changes to JVM Runtime Options . Enterprise Performance Pack uses Unified JVM Logging, which replaces options that print details about the JVM with <code>-Xlog</code> configurations. See Printing JVM Information .
JDK Mission Control (JMC)	You can use JMC for a variety of purposes, such as real-time JMX monitoring. Note that recordings cannot be started from within JMC, however, files that you create from the shell can be opened.
<code>jcmd</code>	Run <code>jcmd <pid> help</code> for the list of available commands.
<code>jdb</code>	Running <code>jdb</code> with the following Serviceability Agent connectors is not supported: <ul style="list-style-type: none"> • <code>SAPIDAttachingConnector</code> • <code>SACoreAttachingConnector</code> • <code>SADebugServerAttachingConnector</code>
<code>jsadebugd</code>	The <code>jsadebugd</code> tool is not supported.
Serviceability Agent scripting	The Serviceability Agent no longer provides JavaScript support.

 **Note:**

- Because Enterprise Performance Pack supports only headless systems, the following tools have been removed from Enterprise Performance Pack: `appletviewer`, `javapackager`, `javaws`, `jconsole`, and `jvisualvm`.
- Enterprise Performance Pack does not include man pages.

The following JDK 8 libraries behave differently or are not supported on Enterprise Performance Pack:

Table 1-3 Libraries That Behave Differently or Are Not Supported on Enterprise Performance Pack

JDK 8 Library	Behavior of Library on Enterprise Performance Pack
Abstract Window Toolkit (AWT)	Enterprise Performance Pack always uses the AWT's headless implementation.
GUI libraries such as JavaFX and Swing	Because Enterprise Performance Pack is for headless systems, GUI libraries aren't supported.

 **Note:**

AWT contains two methods that indicate whether the environment is headless: `GraphicsEnvironment.isHeadless()` and `GraphicsEnvironment.isHeadlessInstance()`. Some AWT methods throw a `HeadlessException` if the environment is headless.

 **Note:**

If you need additional help in running tools and using libraries on Enterprise Performance Pack, contact [My Oracle Support](#).

Other Enterprise Performance Pack Differences

You may experience the following differences when using Enterprise Performance Pack:

Exception Messages

Exception messages that Enterprise Performance Pack prints may be different than those of JDK 8.

Version Strings

The format of Enterprise Performance Pack version strings is different; see [Enterprise Performance Pack Version Strings](#).

Commercial Features

Enterprise Performance Pack doesn't support the following Java SE 8 commercial features:

- Resource Management
- Cooperative Memory Management



Note:

Although Application Class Data Sharing (AppCDS) is a commercial feature in JDK 8, in Enterprise Performance Pack, AppCDS is available as a regular feature.

Consequently, when you specify one of the following command-line options, Enterprise Performance Pack prints a warning message that it's ignored:

- `MemoryRestriction`
- `ResourceManagement`
- `ResourceManagementSampleInterval`

Application Class Data Sharing (AppCDS)

You don't need to specify the `-XX:+UseAppCDS` command-line option to use AppCDS.

You cannot archive classes from the module path; the module system is not part of JDK 8. In addition, you cannot create a dynamic CDS archive file.

2

Changes to JVM Runtime Options

Enterprise Performance Pack includes several runtime options from JDK 17. However, some options from JDK 8 are not available in Enterprise Performance Pack.



Note:

The JDK 8 option `-XX:-UseBiasedLocking` is ignored in Enterprise Performance Pack.

Topics

- [Options Added to Enterprise Performance Pack](#)
- [Options Removed from Enterprise Performance Pack](#)

Options Added to Enterprise Performance Pack

The following runtime options are available in Enterprise Performance Pack but are **not** available in Java SE 8. See the [java command in the JDK 17 documentation](#) for more information about these options:

```
AllocateHeapAt
AsyncLogBufferSize
C1MaxInlineSize
C1MaxTrivialSize
CompactStrings
CompileThresholdScaling
G1UseAdaptiveIHOP
LoopStripMiningIter
LoopStripMiningIterShortLoop
MinHeapSize
NonNMethodCodeHeapSize
NonProfiledCodeHeapSize
PrintFlagsRanges
ProfiledCodeHeapSize
SegmentedCodeCache
SharedArchiveConfigFile
SharedArchiveFile
ShrinkHeapInSteps
StartAggressiveSweepingAt
UseBASE64Intrinsics
UseCMoveUnconditionally
UseDynamicNumberOfCompilerThreads
UseFMA
UseZGC
ZAllocationSpikeTolerance
```

```
ZCollectionInterval  
ZFragmentationLimit  
ZProactive  
ZUncommit  
ZUncommitDelay
```

Options Removed from Enterprise Performance Pack



Note:

The following runtime options are **not** available in Enterprise Performance Pack but are available in Java SE 8. See the [java command in the JDK 8 documentation](#) for more information about these options:

```
AggressiveOpts  
CMSClassUnloadingEnabled  
CMSExpAvgFactor  
CMSIncrementalDutyCycle  
CMSIncrementalDutyCycleMin  
CMSIncrementalMode  
CMSIncrementalOffset  
CMSIncrementalPacing  
CMSIncrementalSafetyFactor  
CMSInitiatingOccupancyFraction  
CMSScavengeBeforeRemark  
CMSTriggerRatio  
CodeCacheMinimumFreeSpace  
PrintAdaptiveSizePolicy  
PrintGCApplicationConcurrentTime  
PrintGCApplicationStoppedTime  
PrintGCDateStamps  
PrintGCTaskTimeStamps  
PrintGCTimeStamps  
PrintStringDeduplicationStatistics  
PrintTenuringDistribution  
UseCMSInitiatingOccupancyOnly  
UseConcMarkSweepGC  
UseParNewGC  
UseParallelOldGC
```

3

Printing JVM Information

Enterprise Performance Pack uses Unified JVM Logging, which replaces JDK 8 options that print details about the JVM with `-Xlog` options.

See [JEP 158: Unified JVM Logging](#) for more information.

The following lists JDK 8 options that are automatically mapped to their equivalent `-Xlog` configuration on Enterprise Performance Pack:

Table 3-1 JDK 8 Options Mapped to Equivalent -Xlog Configuration

JDK 8 Option	-Xlog Configuration
<code>-XX:+PrintGC</code>	<code>-Xlog:gc</code>
<code>-XX:+PrintGCDetails</code>	<code>-Xlog:gc*</code>
<code>-XX:+PrintHeapAtGC</code>	<code>-Xlog:gc+heap=debug</code>
<code>-Xloggc:<filename></code>	<code>-Xlog:gc:<filename></code>

The following table lists JDK 8 options and the `-Xlog` configurations that replace them on Enterprise Performance Pack:

Table 3-2 JDK 8 Options Replaced by Equivalent -Xlog Configuration

JDK 8 Option	-Xlog Configuration
<code>G1PrintHeapRegions</code>	<code>-Xlog:gc+region=trace</code>
<code>G1PrintRegionLivenessInfo</code>	<code>-Xlog:gc+liveness=trace</code>
<code>G1SummarizeConcMark</code>	<code>-Xlog:gc+marking=trace</code>
<code>G1SummarizeRSetStats</code>	<code>-Xlog:gc+remset+exit=trace</code>
<code>GCTraceGCLogFile, NumberOfGCLogFiles, UseGCLogFileRotation</code>	- <code>Xlog:gc*:file=<filename>::filecount=<count>,filesize=<filesize in kb></code>
<code>PrintAdaptiveSizePolicy</code>	<code>-Xlog:gc+ergo*=trace</code>
<code>PrintClassHistogramAfterFullGC</code>	<code>-Xlog:gc+classhisto=trace</code>
<code>PrintClassHistogramBeforeFullGC</code>	<code>-Xlog:gc+classhisto=trace</code>
<code>PrintGCApplicationConcurrentTime</code>	<code>-Xlog:safepoint</code>
<code>PrintGCApplicationStoppedTime</code>	<code>-Xlog:safepoint</code>
<code>PrintGCCause</code>	Always printed
<code>PrintGCDateStamps</code>	<code>-Xlog:gc*:file=<filename>:time,uptime</code>
<code>PrintGCID</code>	Always printed
<code>PrintGCTaskTimeStamps</code>	<code>-Xlog:gc+task+time=debug</code>
<code>PrintGCTimeStamps</code>	The uptime decorator is enabled by default. Alternatively, you can enable it explicitly: <code>-Xlog:gc*:file=<filename>:time,uptime</code>

Table 3-2 (Cont.) JDK 8 Options Replaced by Equivalent -Xlog Configuration

JDK 8 Option	-Xlog Configuration
PrintHeapAtGCExtended	-Xlog:gc+heap=trace
PrintJNIGCStalls	-Xlog:gc+jni=debug
PrintOldPLAB	-Xlog:gc+plab=trace
PrintParallelOldGCPhaseTimes	-Xlog:gc+phases=trace
PrintPLAB	-Xlog:gc+plab=trace
PrintPromotionFailure	-Xlog:gc+promotion=debug
PrintReferenceGC	-Xlog:gc+ref=debug
PrintStringDeduplicationStatistics	-Xlog:gc+stringdedup
PrintTenuringDistribution	-Xlog:gc+age=trace
PrintTLAB	-Xlog:gc+tlab=trace
TraceAdaptiveGCBoundary	No equivalent exists. The functionality of the related option UseAdaptiveGCBoundary has been removed.
TraceDynamicGCThreads	-Xlog:gc+task=trace
TraceMetadataHumongousAllocation	No equivalent exists, but this configuration prints detailed metaspace information: -Xlog:gc+metaspace=trace
G1TraceConcRefinement	-Xlog:gc+refine=debug
G1TraceEagerReclaimHumongousObjects (experimental)	-Xlog:gc+humongous=debug
G1TraceStringSymbolTableScrubbing	No equivalent exists, but this configuration prints detailed string and symbol table information: - Xlog:symboltable*=trace,stringtable*=trace

 **Note:**

Other flags that print information not related to the garbage collector have been mapped to or replaced by an equivalent -Xlog configuration. See [Enable Logging with the JVM Unified Logging Framework](#) in the JDK 17 documentation for the `java` tool for more information.