**JDK 8 兼容性调查**

# 脚本兼容性

## Rhino javascript 引擎已被 Nashorn javascript 引擎取代

**★　参阅 Rhino到Nashorn迁移：**

<https://wiki.openjdk.org/display/Nashorn/Rhino+Migration+Guide>

# JDK 8 和 JDK 7 之间的不兼容性（执行结果）

https://www.oracle.com/java/technologies/javase/8-compatibility-guide.html

## NumberFormat和DecimalFormat

JDK 以前版本的舍入行为在某些极端情况下是错误的，format()当使用非常接近平局的值调用方法时会发生这种错误行为，其中NumberFormatorDecimalFormat实例的模式指定的舍入位置恰好位于平局的位置。在这种情况下，发生了错误的双舍入或错误的非舍入行为。

例：当使用默认推荐的NumberFormatFormatAPI 形式：NumberFormat nf = java.text.NumberFormat.getInstance()后跟 时nf.format(0.8055d)

JDK 7中调用的结果是“ ”的错误输出0.806，而正确的结果是“ 0.805”

## java.util.collections

历史版本Collection.removeAll(Collection)和retainAll(Collection) 会默默地忽略 null 参数，新版本中集合将始终抛出NullPointerException

## 被加入受限制的包

com.sun.media.sound已添加受限包列表中，在 SecurityManager下运行的应用程序将无法访问此包层次结构中的类，除非获得明确许可。该包是一个内部的、不受支持的包，不打算由外部应用程序使用。

com.sun.corba.se及其子包，已添加到 JDK 8 中的受限包列表中、在使用安全管理器运行时不能直接使用

## 随着JAXP 1.6规范更新，XML/JAXP在JDK8有调整

1、This change affects the NodeSet created by DOM Document.

# JDK 8 移除的功能

https://www.oracle.com/java/technologies/javase/8-compatibility-guide.html

## sun.jdbc.odbc库的DBC-ODBC Bridge

从JDK8开始，JDK不再包含 JDBC-ODBC 桥。JDBC-ODBC 桥一直被认为是过渡性的，并且是一种不受支持的产品。

* 一般情项目都不使用的，如果使用了，切换到数据库供应商提供的 JDBC 驱动程序或商业 JDBC 驱动程序

## 移除APT（Annotation-Processing Tool）

## 从JDK中删除了apt工具、及其相关的API和文档。

* 应该对项目的影响不大

# VM垃圾回收（vm/gc）改进：

https://openjdk.org/projects/jdk8/milestones#General\_Availability

* [JEP 122](http://openjdk.java.net/jeps/122)：移除Permanent Generation（永久代）

在 JDK 1.8 中， HotSpot 已经没有 “PermGen space”这个区间了，取而代之是一个叫做 Metaspace（元空间）, JVM的参数：PermSize 和 MaxPermSize 会被忽略并给出警告（如果在启用时设置了这两个参数）。升级到JDK8之后，会发现Java堆 空间有所增长。

　　1、字符串存在永久代中，容易出现性能问题和内存溢出。

　　2、类及方法的信息等比较难确定其大小，因此对于永久代的大小指定比较困难，太小容易出现永久代溢出，太大则容易导致老年代溢出。

　　3、永久代会为 GC 带来不必要的复杂度，并且回收效率偏低。

* [JEP 173](http://openjdk.java.net/jeps/173)：移除一些很少使用的垃圾回收器组合

不推荐使用以下垃圾收集器组合

* DefNew + CMS
* ParNew + SerialOld
* 增量CMS

**原因：**这些组合增加了GC代码库的额外复杂性，消耗了有价值的测试资源，但对用户的价值却很少。为了简化GC代码库,并提高测试和可持续性，建议删除这些组合。

相应的命令行选项会产生警告消息，这些选项将在下一个主要版本之一中删除。

1. 该-Xincgc选项已弃用
2. 该-XX:CMSIncrementalMode选项已弃用。请注意，这也会影响所有CMSIncremental选项。
3. 该-XX:+UseParNewGC选项已弃用，除非您还指定-XX:+UseConcMarkSweepGC.
4. 该-XX:-UseParNewGC选项仅与 结合使用时不推荐使用-XX:+UseConcMarkSweepGC。

# 不推荐使用的API

javac编译器都会生成一条警告消息，除非使用了-nowarn命令行选项。建议修改程序以消除对已弃用 API 的使用。如：'xxxx(java.lang.Object)' 已被弃用

* 再升级过程中根据需要评估是否更新为最新API。

**★　官方提供的不推荐使用API列表**

<https://www.oracle.com/java/technologies/javase/8-compatibility-guide.html#A1000033>