



- [NetBeans IDE](#)
- [Plataforma NetBeans](#)
- [Complementos](#)
- [Documentos y soporte](#)
- [Comunidad](#)
- [Socios](#)



[netbeans.org](#) > [proyectos](#) > [rendimiento](#) > [Sitio web](#)

Ajuste de conmutadores JVM para rendimiento

Las JVM ofrecen una variedad de conmutadores estándar y no estándar que ajustan la asignación de memoria y el comportamiento de recolección de basura. Algunas de estas configuraciones pueden beneficiar el rendimiento de NetBeans.

Tenga en cuenta que los conmutadores JVM -X y especialmente -XX son oficialmente "incompatibles": a menudo son específicos de JVM o JVM. Los conmutadores que se analizan en esta página están disponibles en Sun Microsystems J2SE 1.5.0: los usuarios de otras implementaciones de JVM pueden necesitar eliminar estos conmutadores para ejecutar NetBeans. Por ejemplo, si desea ejecutar el IDE en IBM JDK, debe eliminar `-J-XX:Permsize=32m` y `-J-XX:MaxPermSize=96m` del archivo de configuración `$NB_HOME/etc/netbeans.conf`.

La documentación de Sun's JDK 5 y más reciente se refiere a la [Guía de diagnóstico y solución de problemas](#) que contiene muchos consejos sobre cómo encontrar problemas que a menudo están relacionados con el rendimiento.

Cómo especificar conmutadores JVM

Los conmutadores Java se pasan en la línea de comando que inicia la JVM, por ejemplo, escribiendo `java -jar -Xmx96M someJar.jar`. Dado que NetBeans se inicia mediante un programa de inicio en lugar de llamar directamente a `java`, el iniciador de NetBeans busca estas configuraciones en un archivo especial llamado `netbeans.conf` en el directorio de instalación IDE, etc. Alternativamente, los

[Iniciar sesión](#) | [Únete ahora](#) | [Ayuda](#)

Navegación

- [Casa](#)
- [Mi página](#)
- [Proyectos](#)
- [Personas](#)

Características del proyecto

- [Seguimiento de problemas](#)
- [Bugzilla](#)
- [Listas de correo](#)
- [Descargas](#)
- [Descargas](#)
- [Sitio web](#)
- [Sitio web](#)

Sobre este proyecto

[El rendimiento](#) se inició en noviembre de 2009, es propiedad de [tpavek](#) y tiene [81 miembros](#).

- [» Únete a este proyecto](#)

NetBeans.org

- [Unirse](#)
- [Noticias](#)
- [Lanzamientos y planificación](#)

argumentos se pueden pasar al proceso de Java en la línea de comando anteponiendo `-J` a ellos.

Por ejemplo, para establecer el `-Xmx` (tamaño máximo de `-Xmx` dinámico) para el JVM NetBeans se ejecutará, edite el archivo llamado `netbeans.conf` en el `$NB_HOME/etc` e incluya la opción `-J-Xmx256m` o inicie la ide escribiendo

```
./netbeans -J-Xmx256m
```

o, en sistemas Windows

```
netbeans.exe -J-Xmx256m
```

El archivo `netbeans.conf` puede tener varios conmutadores JVM incluidos en la cadena de texto asignada a la variable `netbeans_default_options`. Tenga en cuenta que Sun JVM no se inicia cuando los interruptores de línea pasados no lo comprende, pero devolverá un mensaje que señala cuál fue el problema, por ejemplo:

```
java -foo
Opción no reconocida: -foo
No se pudo crear la máquina virtual
de Java.
```

- [Listas de correo](#)
- [Bugzilla](#)
- [Contribuir](#)
- [Equipos](#)
- [Pautas](#)

Interruptores generalmente útiles

La siguiente configuración debería producir un rendimiento de configuración mejor que el de fábrica en la mayoría de los sistemas. Con la excepción de establecer el tamaño del "área permanente", estos conmutadores han sido los valores predeterminados para NetBeans durante algún tiempo y ya deberían estar en su archivo `netbeans.conf`.

- **-J-Xms32m** : esta configuración le dice a la máquina virtual Java que establezca su tamaño de almacenamiento dinámico inicial en 32 megabytes. Al decirle a la JVM cuánta memoria debería asignar inicialmente para el montón, lo ahorramos aumentando el montón a medida que NetBeans consume más memoria. Este interruptor mejora el tiempo de inicio. Se usa por defecto en NetBeans, por lo que no necesita especificarlo.
- **-J-Xmx256m** : esta configuración le indica a la máquina virtual Java la cantidad máxima de memoria que debe usar para el

montón. Colocar un límite superior estricto en este número significa que el proceso Java no puede consumir más memoria que la RAM física disponible. Este límite puede aumentarse en sistemas con más memoria. El valor predeterminado actual es 128 MB.

Nota: *No establezca este valor en una cantidad cercana o superior a la cantidad de RAM física en su sistema o provocará un intercambio severo durante el tiempo de ejecución.*

Interruptores más exóticos

A continuación se enumeran algunos conmutadores JVM adicionales que han afectado de manera anecdótica o medible el rendimiento de NetBeans en algunos, no en todos los sistemas. Su millaje puede variar, pero puede valer la pena intentarlo.

- **-J-XX: + UseConcMarkSweepGC o -J-XX: + UseParNewGC** : pruebe estos modificadores si tiene problemas con las pausas intrusivas de recolección de basura. Este modificador hace que la JVM use diferentes algoritmos para los principales eventos de recolección de basura (también para colecciones menores, si se ejecuta en una estación de trabajo multiprocesador), los que no "detienen el mundo" para todo el proceso de recolección de basura. También debe agregar la línea **-J-XX: + CMSClassUnloadingEnabled** y **-J-XX: + CMSPermGenSweepingEnabled** a su archivo `netbeans.conf` para que la descarga de clases esté habilitada (no se usa de manera predeterminada cuando se usa este compilador).
- **-XX: + UseAdaptiveSizePolicy** : este **modificador** puede ayudar a mejorar el rendimiento del recolector de basura y la huella de memoria. Forma parte de la ergonomía del recolector de basura implementada en JDK5.0.
- **-J-XX: + UseParallelGC** : algunas pruebas han demostrado que, al menos en sistemas bastante bien equipados con memoria, la duración de las recolecciones de basura menores se reduce a la mitad cuando se usa este algoritmo de recolección, en sistemas uniprocesadores. Tenga en cuenta que esto es paradójico: este compilador está diseñado para funcionar mejor en sistemas

multiprocesador con montones de gigabytes. No hay datos disponibles sobre su efecto en las principales recolecciones de basura. **Nota:** *este recopilador es mutuamente exclusivo con -J-XX:+UseConcMarkSweepGC* . Las medidas que respaldan el uso de este algoritmo se pueden encontrar en el sitio web de rendimiento .

- **-J-XX: + PrintGCDetails** : se trata de interruptores similares (como **-J-verbose:gc**) que no mejoran el rendimiento, pero proporcionan datos de diagnóstico que muestran información sobre la administración de memoria que son una fuente de entrada útil para el ajuste del rendimiento. Otra forma de obtener estos datos es usar herramientas de monitoreo o perfilador (NetBeans).
- **-J-XX: CompileThreshold = 100** : este interruptor hará que el tiempo de inicio sea más lento, por HotSpot para compilar muchos más métodos hasta el código nativo antes de lo que lo haría de otra manera. El resultado informado es un rendimiento más ágil una vez que se ejecuta el IDE, ya que se compilará más código de la interfaz de usuario en lugar de interpretarlo. Este valor representa la cantidad de veces que se debe llamar a un método antes de que se compile.
- **-J-Djava.net.preferIPv4Stack = true** : este conmutador suprimirá el uso de la pila IPv6 en el código de red y puede evitar pequeños retrasos durante el inicio cuando se resuelve la dirección de entrada. Será útil solo en un sistema donde esté instalado IPv6 pero no esté configurado realmente. *Nota: puede haber otros problemas relacionados con IPv6; consulte, por ejemplo, la discusión sobre la interacción entre fwbuilder y las aplicaciones Java*

Opciones que afectan el comportamiento gráfico

Este documento contiene solo un pequeño subconjunto de interruptores disponibles. Puede encontrar una descripción más detallada sobre las propiedades que afectan el comportamiento de Java 2D (TM) en el documento Propiedades del sistema para la tecnología Java 2D (TM) .

- **-Dsun.java2d.opengl = true** : habilita una

nueva canalización basada en OpenGL para Java 2D que se utiliza para admitir el procesamiento acelerado por hardware con OpenGL. Más detalles sobre esta nueva función JDK5.0 se encuentran en la [documentación de la tecnología Java 2D \(TM\)](#) .

- **-Dsun.java2d.d3d = false** : este **modificador** deshabilita DirectDraw y puede resolver problemas de rendimiento con algunas configuraciones de hardware.
- **-Dawt.nativeDoubleBuffering = true** : este **modificador** hace que Swing asuma que el sistema operativo maneja el doble almacenamiento en búfer y que no debería hacer nada. Esto probablemente no funcionará en una conexión X remota, pero para uso local es muy útil porque literalmente ve que se realiza cada repintado, y hace que sea muy fácil notar si alguna operación está causando repintos gratuitos.
- **El suavizado de fuentes** para widgets Swing se puede activar con la propiedad **-Dswing.aatext = true** . Puede ser útil usarlo junto con la configuración y exportación de la variable de entorno **J2D_PIXMAPS = compartida** al menos en la plataforma Linux para obtener un rendimiento razonable (esto ahora se hace de forma predeterminada en el iniciador (platform5 / lib / nbexec) para que no necesite para configurarlo).

Puede encontrar una lista más exhaustiva de estas opciones en la página de [opciones de JavaTM HotSpot VM](#) que también hace referencia a otro documento que contiene una descripción más detallada de la [optimización del GC](#) .



- [Mapa del sitio](#)
- [Sobre nosotros](#)
- [Contacto](#)
- [Legal y licencias](#)

Al usar este sitio web, usted acepta las Políticas y Términos de Uso de NetBeans (revisión 20160708.bf2ac18). © 2014, Oracle Corporation y / o sus filiales. Patrocinado por **ORACLE**