

Resolviendo problemas de JDK en Eclipse

Cuando inicias Eclipse 4.6.3 haciendo clic en `eclipse.exe` y encuentras el error “Java was started but returned exit code” con una referencia a `Oracle\Java\javapath\javaw.exe`, significa que Eclipse está intentando usar una Máquina Virtual de Java (JVM) que es incompatible o está mal configurada. La ruta `Oracle\Java\javapath\javaw.exe` indica que Eclipse está utilizando una JVM definida en la variable de entorno `PATH` del sistema, que puede no ser adecuada para Eclipse 4.6.3. Para resolver esto, necesitas especificar el Java home correcto para Eclipse configurándolo para usar una JVM compatible. Aquí está cómo hacerlo:

Solución Paso a Paso

1. **Entender el Requisito** Eclipse 4.6.3 (Neon) requiere al menos Java 8 para funcionar correctamente. El error sugiere que la JVM actual (de `Oracle\Java\javapath`) puede no cumplir con este requisito o tiene problemas de configuración. Necesitarás indicar a Eclipse una instalación de Java compatible, como un JDK 8.
2. **Localizar tu Instalación de Java** Identifica dónde está instalada una versión compatible de Java (por ejemplo, JDK 1.8.0) en tu sistema. Ubicaciones comunes en Windows incluyen:
 - `C:\Program Files\Java\jdk1.8.0_XXX` (para Java de 64 bits)
 - `C:\Program Files (x86)\Java\jdk1.8.0_XXX` (para Java de 32 bits) Reemplaza `XXX` con la versión específica de actualización (por ejemplo, `231` para `JDK 1.8.0_231`). Dentro de este directorio, el archivo `javaw.exe` se encuentra en el subdirectorio `bin` (por ejemplo, `C:\Program Files\Java\jdk1.8.0_XXX\bin\javaw.exe`).

Consejo: Para confirmar la versión y la arquitectura, abre una ventana de comandos, navega al directorio `bin` (por ejemplo, `cd C:\Program Files\Java\jdk1.8.0_XXX\bin`), y ejecuta:

```
java -version
```

Busca “64-Bit” o “32-Bit” en la salida para verificar la arquitectura. Asegúrate de que coincida con tu versión de Eclipse (probablemente 64 bits si fue descargada recientemente).

3. **Encontrar el Archivo** `eclipse.ini` El archivo `eclipse.ini` es un archivo de configuración ubicado en el mismo directorio que `eclipse.exe`. Por ejemplo, si Eclipse está instalado en `C:\eclipse`, el archivo estará en `C:\eclipse\eclipse.ini`. Este archivo te permite especificar la JVM que Eclipse debe usar.
4. **Editar el Archivo** `eclipse.ini` Abre `eclipse.ini` en un editor de texto (por ejemplo, Notepad) con privilegios administrativos. Lo modificarás para incluir el argumento `-vm`, que le indica a Eclipse qué JVM usar. Sigue estos pasos:
 - **Revisar el Contenido Existente:** Busca un argumento `-vm`. Si ya está presente, estará seguido de una ruta en la siguiente línea (por ejemplo, `-vm` seguido de `C:/some/path/bin/javaw.exe`). Si apunta a la problemática `Oracle\Java\javapath\javaw.exe`, la reemplazarás. Si no existe un argumento `-vm`, lo agregarás.

- **Agregar o Modificar el Argumento** `-vm`: Inserta las siguientes dos líneas antes de la sección `-vmargs` (si existe) o cerca del inicio del archivo después de los parámetros de inicio iniciales:

```
-vm  
C:/Program Files/Java/jdk1.8.0_XXX/bin/javaw.exe
```

- Usa barras inclinadas (/) en lugar de barras invertidas (\) para evitar problemas de análisis.
- Reemplaza `C:/Program Files/Java/jdk1.8.0_XXX` con la ruta real a tu instalación de Java.

- **Asegurar la Colocación Correcta**: El argumento `-vm` debe aparecer antes de la sección `-vmargs`, que generalmente comienza con `-vmargs` seguido de opciones de JVM como `-Xms256m` o `-Xmx1024m`. Por ejemplo, tu `eclipse.ini` podría verse así después de la edición:

```
-startup  
plugins/org.eclipse.equinox.launcher_1.3.201.v20161025-1711.jar  
--launcher.library  
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.401.v20161122-1740  
-vm  
C:/Program Files/Java/jdk1.8.0_XXX/bin/javaw.exe  
-vmargs  
-Dosgi.requiredJavaVersion=1.8  
-Xms256m  
-Xmx1024m
```

- **Evitar Espacios o Líneas en Blanco Extra**: Asegúrate de que no haya espacios ni líneas en blanco innecesarias inmediatamente después de las líneas `-vm` o de la ruta, ya que esto puede hacer que Eclipse malinterprete la configuración.

5. **Guardar y Probar** Guarda el archivo `eclipse.ini`, luego haz doble clic en `eclipse.exe` para iniciar Eclipse. Si está configurado correctamente, Eclipse debería iniciarse sin el error "Java was started but returned exit code".

Notas Adicionales

- **Coincidencia de Arquitectura**: Asegúrate de que la arquitectura de la JVM (32 bits o 64 bits) coincida con tu instalación de Eclipse. Un Eclipse de 64 bits requiere una JVM de 64 bits, generalmente encontrada en `C:\Program Files\Java`, mientras que un Eclipse de 32 bits necesita una JVM de 32 bits de `C:\Program Files (x86)\Java`.
- **Verificar la Ruta**: Verifica dos veces que `javaw.exe` existe en la ubicación especificada. Si la ruta es incorrecta, Eclipse fallará al iniciar.
- **Opción de Respaldo**: Si esto no funciona, considera establecer la variable de entorno `JAVA_HOME` (por ejemplo, `C:\Program Files\Java\jdk1.8.0_XXX`) y actualizar la ruta del sistema para incluir `%JAVA_HOME%\bin`. Sin embargo, modificar `eclipse.ini` es el método preferido y más confiable para Eclipse.

Al editar el archivo `eclipse.ini` para incluir el argumento `-vm` con la ruta a un `javaw.exe` compatible, especificas efectivamente el Java home para Eclipse, asegurando que use la JVM correcta para iniciar.