

Determine Whether JVM is 32-bit or 64-bit

To resolve the error “Cannot load 64-bit SWT libraries on 32-bit JVM,” you need to ensure that your Java Virtual Machine (JVM) architecture matches the architecture of your Eclipse installation. This error occurs because your Eclipse is trying to use 64-bit SWT libraries, but your JVM is 32-bit, making them incompatible. Below are the steps to check your Java architecture and fix the issue.

Step 1: Check if Your Java is 32-bit or 64-bit

You can determine whether your JVM is 32-bit or 64-bit using one of these methods:

Method 1: Using `java -version`

1. Open a command prompt:

- **Windows:** Press Win + R, type `cmd`, and press Enter.
- **Linux/Mac:** Open your Terminal.

2. Type the following command and press Enter:

```
java -version
```

3. Check the output:

- **64-bit JVM:** The output will include “64-Bit,” for example:

```
java version "1.8.0_311"  
Java(TM) SE Runtime Environment (build 1.8.0_311-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.311-b11, mixed mode)
```

- **32-bit JVM:** The output will not mention “64-Bit,” for example:

```
java version "1.8.0_311"  
Java(TM) SE Runtime Environment (build 1.8.0_311-b11)  
Java HotSpot(TM) Client VM (build 25.311-b11, mixed mode)
```

Method 2: Using `java -d64 -version`

1. In the command prompt, type:

```
java -d64 -version
```

2. Interpret the result:

- **64-bit JVM:** It will display version information like above.
- **32-bit JVM:** You'll see an error, such as:

```
Unrecognized option: -d64
```

```
Error: Could not create the Java Virtual Machine.
```

```
Error: A fatal exception has occurred. Program will exit.
```

If your JVM is 32-bit (as the error suggests), proceed to Step 2 to fix the mismatch.

Step 2: Resolve the Architecture Mismatch

You have two options to resolve this issue:

Option 1: Install a 64-bit JVM (Recommended)

- **Why?** A 64-bit JVM is better for performance and supports larger memory allocations, especially if your system is 64-bit.
- **Steps:**
 1. Download a 64-bit JDK or JRE from a trusted source, such as:
 - Oracle JDK
 - OpenJDK
 2. Install it and note the installation path (e.g., C:\Program Files\Java\jdk1.8.0_XXX\bin\javaw.exe).
 3. Configure Eclipse to use the 64-bit JVM:
 - Open the `eclipse.ini` file in your Eclipse installation folder (e.g., C:\eclipse\eclipse.ini).
 - Add these lines **before** the `-vmargs` section:

```
-vm  
C:/Program Files/Java/jdk1.8.0_XXX/bin/javaw.exe
```
 - Use forward slashes (/) and ensure the path matches your installation.
 4. Save the file and restart Eclipse.

Option 2: Switch to a 32-bit Eclipse

- **Why?** If you want to keep your 32-bit JVM, you can use a 32-bit Eclipse with compatible SWT libraries.
- **Steps:**
 1. Download the 32-bit version of Eclipse from the Eclipse Downloads page.
 2. Install it and launch it with your existing 32-bit JVM.

Key Points to Remember

- **Match Architectures:** Your JVM and Eclipse must have the same architecture:
 - 64-bit Eclipse requires a 64-bit JVM.
 - 32-bit Eclipse requires a 32-bit JVM.
- **Double-Check Paths:** If you edit `eclipse.ini`, ensure the path to `javaw.exe` is correct.
- **System Compatibility:** If your operating system is 64-bit, upgrading to a 64-bit JVM is the better long-term solution.

By following these steps, you'll resolve the error and get Eclipse running smoothly!