

イクライプス JDK の問題を解決する

Eclipse 4.6.3 を `eclipse.exe` をクリックして起動し、エラー“Java was started but returned exit code”が表示され、`Oracle\Java\javapath\javaw.exe` に参照されている場合、Eclipse は互換性がないか、設定が間違っている Java Virtual Machine (JVM) を使用しようとしています。パス `Oracle\Java\javapath\javaw.exe` は、Eclipse がシステムの PATH 環境変数で定義された JVM にデフォルトで依存していることを示しており、これは Eclipse 4.6.3 に適していないかもしれません。これを解決するには、互換性のある JVM を使用するように Eclipse を設定する必要があります。以下にその方法を示します。

ステップバイステップの解決策

- 要件の理解** Eclipse 4.6.3 (Neon) は、正しく動作するために少なくとも Java 8 が必要です。エラーは、現在の JVM (`Oracle\Java\javapath` から) がこの要件を満たしていないか、設定に問題があることを示唆しています。互換性のある Java インストール (例えば、Java 8 JDK) に Eclipse を指す必要があります。
- Java インストールの場所を特定** システム上に互換性のある Java バージョン (例えば、JDK 1.8.0) がインストールされている場所を特定します。Windows の一般的な場所は以下の通りです：
 - `C:\Program Files\Java\jdk1.8.0_XXX` (64 ビット Java の場合)
 - `C:\Program Files (x86)\Java\jdk1.8.0_XXX` (32 ビット Java の場合) `XXX` を特定の更新バージョン (例えば、JDK 1.8.0_231 の場合は 231) に置き換えます。このディレクトリ内の `javaw.exe` ファイルは `bin` サブディレクトリにあります (例えば、`C:\Program Files\Java\jdk1.8.0_XXX\bin\javaw.exe`)。

ヒント: バージョンとアーキテクチャを確認するには、コマンドプロンプトを開き、`bin` ディレクトリに移動 (例えば、`cd C:\Program Files\Java\jdk1.8.0_XXX\bin`) し、以下を実行します：

```
java -version
```

出力に“64-Bit”または“32-Bit”が含まれているか確認して、アーキテクチャを確認します。Eclipse のバージョン (最近ダウンロードした場合はおそらく 64 ビット) と一致していることを確認してください。

- eclipse.ini ファイルの検索** `eclipse.ini` ファイルは、`eclipse.exe` と同じディレクトリにある設定ファイルです。例えば、Eclipse が `C:\eclipse` にインストールされている場合、`eclipse.ini` ファイルは `C:\eclipse\eclipse.ini` にあります。このファイルを使用して、Eclipse が使用する JVM を指定できます。
- eclipse.ini ファイルの編集** 管理者権限でテキストエディタ (例えば、メモ帳) で `eclipse.ini` を開きます。Eclipse が使用する JVM を指定するために `-vm` 引数を追加します。以下の手順に従ってください：
 - 既存の内容の確認:** `-vm` 引数を探します。すでに存在する場合、次の行にパスが続きます (例えば、`-vm` の後に `C:/some/path/bin/javaw.exe`)。問題のある `Oracle\Java\javapath\javaw.exe` を指している場合は、それを置き換えます。`-vm` 引数が存在しない場合は、それを追加します。
 - vm 引数の追加または変更:** `-vmargs` セクション (存在する場合) の前に、またはファイルの上部の初期起動パラメータの後に以下の 2 行を挿入します：

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

- パースの問題を避けるために、スラッシュ (/) を使用します。
- C:/Program Files/Java/jdk1.8.0_XXX を実際の Java インストールパスに置き換えます。

- **適切な配置の確認:** -vm 引数は、通常 -vmargs で始まる JVM オプション（例えば -Xms256m または -Xmx1024m）を含む -vmargs セクションの前に表示される必要があります。編集後、eclipse.ini は以下のようになります：

```
-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
```

- **余分なスペースや空行を避ける:** -vm またはパス行の直後に余分なスペースや空行がないことを確認してください。そうしないと、Eclipse が設定を誤解釈する可能性があります。

5. **保存とテスト** eclipse.ini ファイルを保存し、eclipse.exe をダブルクリックして Eclipse を起動します。正しく設定されている場合、Eclipse は“Java was started but returned exit code”エラーなしで起動します。

追加の注意点

- **アーキテクチャの一致:** JVM のアーキテクチャ（32 ビットまたは 64 ビット）が Eclipse インストールと一致していることを確認してください。64 ビットの Eclipse は、通常 C:\Program Files\Java にある 64 ビットの JVM を必要とし、32 ビットの Eclipse は C:\Program Files (x86)\Java にある 32 ビットの JVM を必要とします。
- **パスの確認:** 指定された場所に javaw.exe が存在することを確認してください。パスが間違っている場合、Eclipse は起動できません。
- **フォールバック オプション:** これが機能しない場合は、JAVA_HOME 環境変数（例えば、C:\Program Files\Java\jdk1.8.0_XXX）を設定し、システムの PATH に %JAVA_HOME%\bin を追加することを検討してください。ただし、eclipse.ini を編集する方法が Eclipse にとってより優れた方法です。

eclipse.ini ファイルに -vm 引数と互換性のある javaw.exe のパスを含めることで、Eclipse が正しい JVM を使用するように指定し、Eclipse が正しく起動することを確保します。