

□□□□□□ □□ समस्या समाधान

जब आप □□□□□□ 4.6.3 को eclipse.exe पर क्लिक करके शुरू करते हैं और “□□□□ □□ □□□□□□ □□ □□□□□□ □□□□ □□□□” के साथ एक त्रुटि का सामना करते हैं, जिसमें Oracle\Java\javapath\javaw.exe का संदर्भ है, तो यह मतलब है कि □□□□□□ एक □□□□ □□□□□□ □□□□□□ (□□□□) का उपयोग करने की कोशिश कर रहा है जो या तो अनुपयुक्त है या गलत ढंग से कॉन्फ़िगर किया गया है। पथ Oracle\Java\javapath\javaw.exe यह संकेत देता है कि □□□□□□ आपके सिस्टम के □□□□ पर्यावरण चर को डिफ़ॉल्ट में परिभाषित किए गए □□□□ का उपयोग कर रहा है, जो □□□□□□ 4.6.3 के लिए उपयुक्त हो सकता है। इस समस्या को सुलझाने के लिए, आपको □□□□□□ को एक अनुकूल □□□□ का उपयोग करने के लिए कॉन्फ़िगर करने की आवश्यकता है। यहाँ कैसे करना है:

चरण-दर-चरण समाधान

1. **आवश्यकता समझें** □□□□□□ 4.6.3 (□□□□□) को सही तरह से चलाने के लिए कम से कम □□□□ 8 की आवश्यकता होती है। त्रुटि यह सुझाव देती है कि वर्तमान □□□□ (Oracle\Java\javapath से) इस आवश्यकता को पूरा नहीं कर सकता या इसमें कॉन्फ़िगरेशन समस्याएं हैं। आपको □□□□□□ को एक अनुकूल □□□□ इंस्टॉलेशन (जैसे □□□□ 8 □□□□) के साथ जोड़ने की आवश्यकता होगी।
2. **अपने □□□□ इंस्टॉलेशन को खोजें** अपने सिस्टम पर एक अनुकूल □□□□ संस्करण (जैसे □□□□ 1.8.0) का पता लगाएं। □□□□□□ पर आम स्थान हैं:

□ C:\Program Files\Java\jdk1.8.0_XXX (64-बिट □□□□ के लिए)

□ C:\Program Files (x86)\Java\jdk1.8.0_XXX (32-बिट □□□□ के लिए) XXX को विशेष अपडेट संस्करण (जैसे 231 के लिए □□□□ 1.8.0_231) से बदलें। इस डायरेक्टरी के अंदर, javaw.exe फ़ाइल bin सबडायरेक्टरी में स्थित है (जैसे, C:\Program Files\Java\jdk1.8.0_XXX\bin\javaw.exe)।

टिप: संस्करण और आर्किटेक्चर की पुष्टि करने के लिए, एक कमांड प्रॉम्प्ट खोलें, bin डायरेक्टरी में जाएं (जैसे, cd C:\Program Files\Java\jdk1.8.0_XXX\bin), और निम्नलिखित को चलाएं:

```
java -version
```

आउटपुट में “64-□□□□” या “32-□□□□” देखें ताकि आर्किटेक्चर की पुष्टि हो सके। सुनिश्चित करें कि यह आपके □□□□□□□□ संस्करण (अगर हाल ही में डाउनलोड किया गया है तो संभवतः 64-बिट) से मेल खाता है।

3. **eclipse.ini फ़ाइल खोजें** eclipse.ini फ़ाइल एक कॉन्फ़िगरेशन फ़ाइल है जो eclipse.exe के साथ उसी डायरेक्टरी में स्थित है। उदाहरण के लिए, अगर □□□□□□□□ C:\eclipse में इंस्टॉल किया गया है, तो फ़ाइल C:\eclipse\eclipse.ini पर होगी। इस फ़ाइल के माध्यम से आप □□□□□□□□ को कौन सा □□□□ का उपयोग करना चाहिए, यह निर्दिष्ट कर सकते हैं।
4. **eclipse.ini फ़ाइल को संपादित करें** एक टेक्स्ट एडिटर (जैसे, नोटपैड) में eclipse.ini को प्रशासनिक अधिकार के साथ खोलें। आप इसे -vm आर्ग्यूमेंट को जोड़ने या संपादित करने के लिए बदलेंगे, जो □□□□□□□□ को कौन सा □□□□ का उपयोग करना चाहिए, यह बताता है। निम्नलिखित चरणों का पालन करें:

□ **वर्तमान सामग्री की जांच करें:** -vm आर्ग्यूमेंट की तलाश करें। अगर यह पहले से मौजूद है, तो यह अगले पंक्ति पर एक पथ के साथ होगा (जैसे, -vm के बाद C:/some/path/bin/javaw.exe)। अगर यह गलत Oracle\Java\javapath\javaw.exe को संकेत करता है, तो आप इसे बदल देंगे। अगर कोई -vm आर्ग्यूमेंट मौजूद नहीं है, तो आप इसे जोड़ेंगे।

□ **-vm आर्ग्यूमेंट को जोड़ें या संपादित करें:** निम्नलिखित दो पंक्तियों को -vmargs खंड (अगर मौजूद है) से पहले या फ़ाइल के शीर्ष पर प्रारंभिक स्टार्टअप पैरामीटरों के बाद डालें:

-vm

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

- पर्सिंग समस्याओं से बचने के लिए, फॉरवर्ड स्लैश (/) का उपयोग करें, बजाय बैकस्लैश (\).
- C:/Program Files/Java/jdk1.8.0_XXX को अपने `%%~dp0` इंस्टॉलेशन के वास्तविक पथ से बदलें।

□ **सही स्थान सुनिश्चित करें:** -vm आर्ग्यूमेंट -vmargs खंड से पहले होना चाहिए, जो आमतौर पर -vmargs के साथ शुरू होता है, फिर `%%~dp0` विकल्प जैसे -Xms256m या -Xmx1024m। उदाहरण के लिए, संपादित करने के बाद आपका eclipse.ini इस तरह दिख सकता है:

-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

□ **अनावश्यक स्पेस या खाली पंक्तियों से बचें:** सुनिश्चित करें कि -vm या पथ पंक्तियों के तुरंत बाद कोई अनावश्यक स्पेस या खाली पंक्तियां नहीं हैं, क्योंकि इससे `%%~dp0` को कॉन्फिगरेशन को गलत समझने की संभावना हो सकती है।

5. **सहेजें और परीक्षण करें** eclipse.ini फ़ाइल को सहेजें, फिर eclipse.exe पर दबाएं ताकि `%%~dp0` शुरू हो सके। अगर सही तरह से कॉन्फिगर किया गया है, तो `%%~dp0` "%%~dp0" त्रुटि के बिना शुरू हो जाएगा।

अतिरिक्त टिप्पणियाँ

- **आर्किटेक्चर मेल खाएं:** सुनिश्चित करें कि `%%~dp0` आर्किटेक्चर (32-बिट या 64-बिट) आपके `%%~dp0` इंस्टॉलेशन से मेल खाता है। एक 64-बिट `%%~dp0` को एक 64-बिट `%%~dp0` की आवश्यकता होती है, जो आमतौर पर C:\Program Files\Java में पाया जाता है, जबकि एक 32-बिट `%%~dp0` को एक 32-बिट `%%~dp0` की आवश्यकता होती है, जो C:\Program Files (x86)\Java से मिलता है।
- **पथ की पुष्टि करें:** सुनिश्चित करें कि javaw.exe निर्दिष्ट स्थान पर मौजूद है। अगर पथ गलत है, तो `%%~dp0` शुरू नहीं होगा।
- **फॉलबैक विकल्प:** अगर यह काम नहीं करता, तो JAVA_HOME पर्यावरण चर (जैसे C:\Program Files\Java\jdk1.8.0_XXX) को सेट करने और सिस्टम `%%~dp0` को %JAVA_HOME%\bin को शामिल करने का विचार करें। हालांकि, eclipse.ini को संपादित करना `%%~dp0` के लिए अधिक पसंदीदा और विश्वसनीय तरीका है।

eclipse.ini फ़ाइल को संपादित करके -vm आर्ग्यूमेंट के साथ एक अनुकूल javaw.exe के पथ को जोड़ने से, आप `%%~dp0` के लिए `%%~dp0` होम को स्पष्ट रूप से निर्दिष्ट करते हैं, सुनिश्चित करते हैं कि यह सही `%%~dp0` का उपयोग करके शुरू हो।