

# 【开发坑】 new File(path).exists()和 Files.exists(Path) 返回结果不一致问题

作者: [yexuejc](#)

原文链接: <https://ld246.com/article/1682061018126>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

文件存在的情况下，如：D:/doc.txt

使用`new File("D:/doc.txt").exists()` 返回的结果为`true`,表示文件存在

使用`Files.exists(Path.of("D:/doc.txt"))`返回的结果为`false`,文件不存在?

追溯原因：

在[Oracle Java Bug Databse](#)上发现了有人在2004年的1.4.2版本的Java就发现了相同的问题，他看源码后发现是因为`File.exists`其实是用了Linux的`stat(2)`命令，如果这个命令失败了，`File.exists`就认为是文件不存在。

#### A DESCRIPTION OF THE PROBLEM :

When operating on NFS-mounted volumes, `java.io.File.exists` will sometimes return "false" ven though the file referenced actually does exist.

The actual problem is a stale NFS file handle, which can occur for a number of reasons.

An examination of the java source code shows that `File.exists` ends up performing a `stat(2)`. If he `stat` fails, then `File.exists` assumes that the file being `stat`' ed does not exist. This is a false ssumption. `stat(2)` can fail for several reasons other than `ENOENT`. One such "other" error is `ESTALE`, which is what I get if I use JNI to perform a "native" `stat(2)` on the same file that `File exists` claims does not exist.

#### STEPS TO FOLLOW TO REPRODUCE THE PROBLEM :

I do not have a simple method for reproducing this problem, however an examination of the ava sources will bear out my explantion above.

#### EXPECTED VERSUS ACTUAL BEHAVIOR :

EXPECTED -

If the file exists, I expect `java.io.File.exists` to return "true"

ACTUAL -

`java.io.File.exists` returns "false" occasionally.

#### REPRODUCIBILITY :

This bug can be reproduced occasionally.

(Incident Review ID: 191975)

## 然后解决方案是不解决。。。。

#### EVALUATION

The error code should be examined from the `stat`. But `exists` does not throw `IOException` so w at to do in such a case would have to be determined.

###@###.### 2004-03-24

Submitter is correct but this API is too limited to do anything useful when the `stat` returns a r al error. We will try to address this issue in JSR203.

###@###.### 2004-03-25

## 所有只有换个实现方法了。检查文件是否存在时，可能会出现三种结果：

- 该文件存在。

- 该文件不存在。
- 该文件的状态未知，因为该程序无权访问该文件。

有几种方法可以在 Java 中检查文件是否存在。如果文件存在，返回 `true`；文件不存在或文件的状态未知时，返回 `false`。

## 一、使用 `File.exists()` 方法

思路是使用该 `File.exists()` 方法判断指定路径名表示的文件是否存在。如果文件存在，该方法返回 `true` 否则为 `false`。

请注意，`File.exists()` 当您的路径指向目录时返回 `true`。因此，建议将此方法与 `File.isDirectory()` 检查目录的方法一起调用。如下所示：

```
import java.io.File;

class Main
{
    // Method to check if the file exists and is not a directory
    public static boolean isFileExists(File file) {
        return file.exists() && !file.isDirectory();
    }

    public static void main(String[] args)
    {
        String filePath = "C:\\doc.txt";
        File file = new File(filePath);

        if (isFileExists(file)) {
            System.out.println("File exists!!");
        }
        else {
            System.out.println("File doesn't exist or program doesn't have access " +
                "to the file");
        }
    }
}
```

请注意，在 NFS 挂载的卷上操作时，`java.io.File.exists` 有时会返回 `false`，即使引用的文件确实存在。[在此处](#) 查看错误详细信息。

## 二、使用 `File.isFile()` 方法

我们已经看到，`File.exists()` 如果您的路径指向一个目录，它会返回 `true`。要明确避免检查目录，建议使用 `File.isFile()` method 而不是 `File.exists()` method。该 `File.isFile()` 方法测试指定路径所指的文件是否为普通文件，即文件是否为目录。

```
import java.io.File;

class Main
{
    // Method to check if the file exists
```

```

public static boolean isFileExists(File file) {
    return file.isFile();
}

public static void main(String[] args)
{
    String filePath = "C:\\doc.txt";
    File file = new File(filePath);

    if (isFileExists(file)) {
        System.out.println("File exists!!");
    }
    else {
        System.out.println("File doesn't exist or program doesn't have access " +
            "to the file");
    }
}
}

```

### 三.使用NIO Files.exists()

从 Java 7 开始，我们可以使用[java.nio.file.Files](#)，它提供了几个对文件、目录或其他类型的文件进行操作的静态方法。要简单地检查文件是否存在，我们可以使用[exists\(\)](#)类[notExists\(\)](#)的方法[java.nio.file.Files](#)。

[exists\(\)](#)如果文件存在，则该方法返回 true，而[notExists\(\)](#)当文件不存在时，该方法返回 true。如果[exists\(\)](#)和[notExists\(\)](#)都返回false，则无法验证文件是否存在。当程序无权访问该文件时，可能会发生这种情况。

请注意，[Files.exists\(\)](#)当您的路径指向目录时返回 true。因此，建议将此方法与检查目录文件的方法[Files.isDirectory\(\)](#)一起使用。如下所示：

```

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

class Main
{
    public static void main(String[] args)
    {
        String filePath = "C:\\doc.txt";

        Path path = Paths.get(filePath);

        boolean exists = Files.exists(path);
        boolean notExists = Files.notExists(path);
        boolean isDir = Files.isDirectory(path);

        if (isDir) {
            System.out.println("File is a Directory");
        }
        else if (exists) {
            System.out.println("File exists!!");
        }
    }
}

```

```
    else if (notExists) {
        System.out.println("File doesn't exist!!");
    }
    else {
        System.out.println("Program doesn't have access to the file!!");
    }
}
}
```

或者，我们可以使用以下方法检查路径是否为常规文件（而不是目录）`Files.isRegularFile()`：

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

class Main
{
    public static void main(String[] args)
    {
        String filePath = "C:\\doc.txt";

        Path path = Paths.get(filePath);
        boolean exists = Files.isRegularFile(path);

        if (exists) {
            System.out.println("File exists!!");
        }
        else {
            System.out.println("File doesn't exist!!");
        }
    }
}
```

参考：

[java.io.File.exists在文件存在时返回false的BUG](#)

[Check if a file exists in Java](#)

[Java Bug Database](#)