

Java Copy File

Java Copy File

To copy file in Java from one location to another location, there are many ways. In this tutorial, we will go through some of the ways: 1. Using Input and Output Streams, 2. Using Files.copy, and 3. Using FileUtils class of apache's commons.io java package.

Example 1 – Java Copy File with Streams

In this example, we are going to use FileInputStream and FileOutputStream. We shall follow the below sequence of steps to copy file using Streams.

1. Prepare source and destination file objects.
2. Create FileInputStream for source file object and FileOutputStream for destination file object.
3. Read bytes from FileInputStream and then write those bytes to FileOutputStream. Repeat until you write all the content from source to destination.

CopyFile.java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CopyFile {
    public static void main(String[] args) throws IOException {
        File source = new File("files/tutorials.png");
        File destination = new File("files/1/tutorials_copy.png");

        try (FileInputStream fis = new FileInputStream(source);
            FileOutputStream fos = new FileOutputStream(destination)) {
            byte[] buffer = new byte[1024];
            int length;
            while ((length = fis.read(buffer)) > 0) {
                fos.write(buffer, 0, length);
            }
            System.out.println("File successfully copied.");
        } catch (IOException e) {
            System.out.println("Handle the exception");
            e.printStackTrace();
        }
    }
}
```

Run the program and you should see the source file copies to the destination.

We have used Java – Try with resource in the above program.

Creating FileInputStream and FileOutputStream with the file objects could throw IOException. Handle the exception based on your application requirement.

Example 2 – Java Copy File with Files class

In this example, we shall use Files class of java.nio.file package. Files.copy() method takes two mandatory arguments for source and destination paths. In addition to those, copy() method takes an optional third argument where we can provide options for copying.

The options parameter takes one of the following values:

- `StandardCopyOption.REPLACE_EXISTING` – replaces destination if it already exists.
- `StandardCopyOption.COPY_ATTRIBUTES` – an attempt is made to copy the file attributes associated with source file to the destination file.
- `StandardCopyOption.ATOMIC_MOVE` – moves the file from source to destination.
- `StandardCopyOption.NOFOLLOW_LINKS` – symbolic links are not followed.

In this example, we used the options string `StandardCopyOption.REPLACE_EXISTING`.

CopyFile.java

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;

public class CopyFile {

    public static void main(String[] args) throws IOException {

        File source = new File("files/tutorials.png");
        File destination = new File("files/1/tutorials_copy.png");

        Files.copy(source.toPath(), destination.toPath(),
            StandardCopyOption.REPLACE_EXISTING);
    }
}
```

Run the program and you should see the source file copies to the destination.

Example 3 – Java Copy File using org.apache.commons.io.FileUtils

In this example, we shall use FileUtils class of commons.io package. FileUtils.copyFile() copies file from source to destination. The source and destination files are passed as arguments to copyFile() function respectively.

CopyFile.java

```
import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FileUtils;

public class CopyFile {

    public static void main(String[] args) throws IOException {

        File source = new File("files/tutorials.png");
        File destination = new File("files/1/tutorials_copy.png");

        FileUtils.copyFile(source, destination);
    }
}
```

Run the program and the source file shall be copied to destination path. If the file specified by destination is already present, then it shall be overridden by the source file.

Conclusion

Concluding this [Java Tutorial](#), we learned how to copy a file from one location to another using different methods.

Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)
- ◆ [Java Datatypes](#)
- ◆ [Java Variable Types](#)
- ◆ [Java Access Modifiers](#)
- ◆ [Java Operators](#)

◆ [Java Decision Making](#)

◆ [Java Loops](#)

◆ [Java Array](#)

◆ [Java OOPs](#)

◆ [Java String](#)

◆ [Java Exception Handling](#)

◆ [Java File Operations](#)

◆ [Java Date & Time](#)

◆ [Java MySQL](#)

◆ [Java Random](#)

◆ [Java Math](#)