How to append text into File in Java - FileWriter Example

Some times we need to append text into File in Java instead of creating new File. Thankfully Java File API is very rich and it provides several ways to append text into File in Java. Previously we have seen how to create file and directory in Java and how to read and write to text file in Java and in this Java IO tutorial we will see how to append text into file in Java. We are going to use standard `FileWriter` and `BufferedWriter` approach to append text to File. One of the key point to remember while using `FileWriter` in Java is to initialize `FileWriter` to append text i.e. writing bytes at the end of File rather than writing on beginning of the File. In next section we will see complete *Java program to append text into File in Java*.

By the way you can also use `FileOutputStream` instead of `FileWriter` if you would like to write bytes, instead of text. Similar to `FileWriter`, `FileOutputStream` constructor also takes a boolean append argument to open connection to append bytes into File in Java.

Java program to append text to File in Java using FileWriter

In this section we will see a fully functional Java program which appends text at the end of File. In this program, we have used `FileWriter` and initialized it will append as `true`. For better performance we have wrapped `FileWriter` into a `BufferedWriter`, Similar to wrapping `FileInputStream` into `BufferedReader` to read File line by line in Java .

Since this is a test program we have not finally block to close connection and also instead of handling Exception we have thrown them to keep the code simple and readable. In production code, you should always close File connection on finally block and ready to handle `FileNotFoundException` and `IOException` in Java.

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;

/**
 * Simple Java program to append content and text into File.
 * You can either use byte stream approach or character reader approach to append
 * text to File. Readers will be faster than Stream and its advised to use BufferedWriter
 * for better performance. Exception from code are thrown to improve clarity of code,
 * In real word you should handle them properly.
```

Interview Questions

- core java interview question (161)
- data structure and algorithm (45)
- Coding Interview Question (32)
- SQL Interview Questions (24)
- thread interview questions (20)
- database interview questions (18)
- servlet interview questions (17)
- collections interview questions (15)
- spring interview questions (9)
- Programming interview question (4)
- hibernate interview questions (4)

Translate this blog

Select Language

Powered by Google Translate

```
 *
 * @author Javin Paul
 */
public class FileAppendTest{



    public static void main(String args[]) throws FileNotFoundException, IOException {
        //name of File on which text will be appended,
        //currently file contains only one line
        //as "This data is before any text appended into file."
        String path = "C:/sample.txt";

        //creating file object from given path
        File file = new File(path);

        //FileWriter second argument is for append if its true than FileWritter will
        //write bytes at the end of File (append) rather than beginning of file
        FileWriter fileWriter = new FileWriter(file,true);

        //Use BufferedWriter instead of FileWriter for better performance
        BufferedWriter bufferFileWriter  = new BufferedWriter(fileWriter);
        fileWriter.append("This text should be appended in File form Java Program");

        //Don't forget to close Streams or Reader to free FileDescriptor associated with it
        bufferFileWriter.close();

        System.out.println("Java Program for appending content into File has been completed");

    }

}
```

Output:

```
Original File Content: This data is before any text appended into file.
Modified File Content: This data is before any text appended into file.This text should be
appended in File form Java Program  You can also append contents or bytes by using
FileOutputStream, FileOutputStream(String path, boolean append) takes a boolean parameter to
append into File, which will ensure that new bytes will be written at the end of File rather
than at beginning of File.
```

That's all on **how to append text to File in Java**. It's rather simple Java program with only one thing to keep in mind, initializing `FileWriter` with boolean append as `true`, so that `FileWriter` writes content at the end of file instead start of the File.

Other **Java IO tutorials** from Javarevisited Blog

How to read XML files in Java using DOM parser

How to find hidden files in Java

What is Memory Mapped File in Java with Example

5 ways to convert InputStream to String in Java

How to read input from command line using Scanner in Java

Java Tutorials

- date and time tutorial (18)
- FIX protocol tutorial (16)
- java collection tutorial (53)
- java IO tutorial (25)
- Java JSON tutorial (6)
- Java multithreading Tutorials (33)
- Java Programming Tutorials (27)
- Java xml tutorial (9)

Search This Blog

Search