

Append to a file in java using BufferedWriter, PrintWriter, FileWriter

BY CHAITANYA SINGH | FILED UNDER: [JAVA I/O](#)

In this tutorial we will learn how to append content to a file in Java. There are two ways to append:

- 1) Using `FileWriter` and `BufferedWriter`: In this approach we will be having the content in one of more Strings and we will be appending those Strings to the file. The file can be appended using `FileWriter` alone however using `BufferedWriter` improves the performance as it maintains a buffer.
- 2) Using `PrintWriter`: This is one of best way to append content to a file. Whatever you write using `PrintWriter` object would be appended to the File.

1) Append content to File using FileWriter and BufferedWriter

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

class AppendFileDemo
{
    public static void main( String[] args )
    {
        try{
            String content = "This is my content which would be appended " +
                             "at the end of the specified file";
            //Specify the file name and path here
            File file =new File("C://myfile.txt");

            /* This logic is to create the file if the
             * file is not already present
             */
            if(!file.exists()){
                file.createNewFile();
            }

            //Here true is to append the content to file
            FileWriter fw = new FileWriter(file,true);
            //BufferedWriter writer give better performance
            BufferedWriter bw = new BufferedWriter(fw);
            bw.write(content);
            //Closing BufferedWriter Stream
            bw.close();

            System.out.println("Data successfully appended at the end of file");
        }catch(IOException ioe){
```

```

        System.out.println("Exception occurred:");
        ioe.printStackTrace();
    }
}
}

```

Output:

Data successfully appended at the end of file
 Lets say myfile.txt content was:

This is the already present content of my file
 After running the above program the content would be:

This is the already present content of my file This is my content which
 would be appended at the end of the specified file

2) Append content to File using PrintWriter

PrintWriter gives you more flexibility. Using this you can easily format the content which is to be appended to the File.

```

import java.io.File;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.IOException;

class AppendFileDemo2
{
    public static void main( String[] args )
    {
        try{
            File file =new File("C://myfile.txt");
            if(!file.exists()){
                file.createNewFile();
            }
            FileWriter fw = new FileWriter(file,true);
            BufferedWriter bw = new BufferedWriter(fw);
            PrintWriter pw = new PrintWriter(bw);
            //This will add a new line to the file content
            pw.println("");
            /* Below three statements would add three
             * mentioned Strings to the file in new lines.
             */
            pw.println("This is first line");
            pw.println("This is the second line");
            pw.println("This is third line");
            pw.close();

            System.out.println("Data successfully appended at the end of file");

        }catch(IOException ioe){
            System.out.println("Exception occurred:");
            ioe.printStackTrace();
        }
    }
}

```

```
}  
}
```

Output:

Data successfully appended at the end of file

Lets say myfile.txt content was:

This is the already present content of my file

After running the above program the content would be:

This is the already present content of my file

This is first line

This is the second line

This is third line