

JAVA.IO.DATAOUTPUTSTREAM.FLUSH METHOD

http://www.tutorialspoint.com/java/io/dataoutputstream_flush.htm

Copyright © tutorialspoint.com

Description

The **java.io.DataOutputStream.flush** method forces bytes to be written to the underlying stream.

Declaration

Following is the declaration for **java.io.DataOutputStream.flush** method:

```
public void flush()
```

Parameters

- **NA**

Return Value

This method does not return any value.

Exception

- **IOException** -- If an I/O error occurs.

Example

The following example shows the usage of java.io.DataOutputStream.flush method.

```
package com.tutorialspoint;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;

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

        InputStream is = null;
        DataInputStream dis = null;
        FileOutputStream fos = null;
        DataOutputStream dos = null;
        String[] s = {"Hello", "World!!"};

        try{
            // create file output stream
            fos = new FileOutputStream("c:\\test.txt");

            // create data output stream
            dos = new DataOutputStream(fos);

            // for each string in string buffer
            for(String j:s)
            {
                // write string encoded as modified UTF-8
                dos.writeUTF(j);
            }

            // forces data to the underlying file output stream
            dos.flush();
        }
    }
}
```

```

// create file input stream
is = new FileInputStream("c:\\test.txt");

// create new data input stream
dis = new DataInputStream(is);

// available stream to be read
while(dis.available()>0)
{
    // reads characters encoded with modified UTF-8
    String k = dis.readUTF();

    // print
    System.out.print(k+" ");
}
}catch(Exception e){
    // if any error occurs
    e.printStackTrace();
}finally{

    // releases all system resources from the streams
    if(is!=null)
        is.close();
    if(dis!=null)
        dis.close();
    if(fos!=null)
        fos.close();
    if(dos!=null)
        dos.close();
}
}
}

```

Let us compile and run the above program, this will produce the following result:

Hello World!!

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js