<http://javarevisited.blogspot.com.by/2011/12/read-and-write-text-file-java.html>

How to read and write in text file in Java

Java has excellent support for **reading from file and writing to file in Java**. In last post we have seen [how to create file and directory in Java](http://javarevisited.blogspot.com/2011/12/create-file-directory-java-example.html) and now we will see how to read content from file in java and how we will write text into file in Java. File provides persistent solution to Java developer; in almost every Java application you need to store some of the data in persistent may be its user configuration, system configuration or related to state of application but without persistence no enterprise Java application can buildup. Normally Database is preferred choice for storing data persistently but in **high frequency trading** one doesn’t have liberty to afford network latency to travel database and latency introduced by database itself, though database still offer best choice for historical data , transactional details related to orders and trades are commonly stored in In Memory files provided by Java.nio API.

## How to read and write from text file in Java

[read and writ from text file in Java](http://javarevisited.blogspot.com/2011/08/classnotfoundexception-in-java-example.html)Though in this *java file tutorial* we are not talking about In Memory files, we will discuss plain old File classes and **how to read and write into File in Java** by using java.io package classes.

If we are working on **Standalone application** then we have access to local file system and we can easily using the **java API** read and write on files, but we if we our application is running on browser based system then this will not work. If we are using input and output stream for reading and writing it’s very easy to understand. We have to follow three simple steps to achieve this task.

       First get the **File** object

       Create the **File Stream** from File object

       Use this File Stream for reading or writing the data to the file in the file system.

Apart from this we need to remember some points which should be taken care at the time of reading and writing to the file in Java.

       Always try to use reference of **abstract classes** or interfaces in place of implemented class or we can say concrete class it is a one of the good java practice also.

       If needed use **buffering**  it’s a good practice because calling a **read()** method for single byte **JVM** will call the native operating system method and  calling a operating system method is expensive so Buffering will reduce this overhead from some extent.

       If using buffer then mentions the buffer size also it will affect the read time and CPU time also.

       Always Handle the Exceptions (**IOException and FileNotFoundException)**

       Don’t forget to call **close()** method for resource which we are using such as File or Directory in Java. You can also use [automatic resource management in JDK7](http://javarevisited.blogspot.com/2011/09/arm-automatic-resource-management-in.html) to automatically close any open resource in Java.

### File read and write Example in Java

Now we see simple example of reading and writing a **binary data to a file** in Java.

**class** FileStreamsReadnWrite {

**public** **static** **void** main(String[] args) {

**try** {

                     File stockInputFile = **new** File("C://stock/stockIn.txt");

                     File StockOutputFile = **new** File("C://stock/StockOut.txt");

                     /\*

                      \* Constructor of FileInputStream throws FileNotFoundException if

                      \* the argument File does not exist.

                      \*/

                     FileInputStream fis = **new** FileInputStream(stockInputFile);

                     FileOutputStream fos = **new** FileOutputStream(StockOutputFile);

**int** count;

**while** ((count = fis.read()) != -1) {

                           fos.write(count);

                     }

                     fis.close();

                     fos.close();

              } **catch** (FileNotFoundException e) {

                     System.*err*.println("FileStreamsReadnWrite: " + e);

              } **catch** (IOException e) {

                     System.*err*.println("FileStreamsReadnWrite: " + e);

              }

       }

}

This is very simple example where we are reading from stockInputfile and whatever contents are there we are writing that to stockoutput file one important point here is that FileInputStream should not be used to read character data file.It is meant for reading binary data such as an image file.

That’s all on *how to read from file in Java* and *writing data into file in Java*. It’s very important to understand java.io package to get mastery in java and best way it to write examples and simple program and understand concept.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*