

JAVA.IO.OBJECTOUTPUTSTREAM.WRITEUNSHARED METHOD

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

Copyright © tutorialspoint.com

Description

The **java.io.ObjectOutputStream.writeUnshared** method writes an "unshared" object to the ObjectOutputStream. This method is identical to `writeObject`, except that it always writes the given object as a new, unique object in the stream

as opposed to a back – reference pointing to a previously serialized instance. Specifically:

- An object written via `writeUnshared` is always serialized in the same manner as a newly appearing object *an object that has not been written to the stream yet*, regardless of whether or not the object has been written previously.
- If `writeObject` is used to write an object that has been previously written with `writeUnshared`, the previous `writeUnshared` operation is treated as if it were a write of a separate object. In other words, ObjectOutputStream will never generate back-references to object data written by calls to `writeUnshared`.

While writing an object via `writeUnshared` does not in itself guarantee a unique reference to the object when it is deserialized, it allows a single object to be defined multiple times in a stream, so that multiple calls to `readUnshared` by the receiver will not conflict. Note that the rules described above only apply to the base-level object written with `writeUnshared`, and not to any transitively referenced sub-objects in the object graph to be serialized.

ObjectOutputStream subclasses which override this method can only be constructed in security contexts possessing the "enableSubclassImplementation" SerializablePermission; any attempt to instantiate such a subclass without this permission will cause a SecurityException to be thrown.

Declaration

Following is the declaration for **java.io.ObjectOutputStream.writeUnshared** method.

```
public void writeUnshared(Object obj)
```

Parameters

- **obj** -- object to write to stream

Return Value

This method does not return a value.

Exception

- **NotSerializableException** -- if an object in the graph to be serialized does not implement the Serializable interface
- **InvalidClassException** -- if a problem exists with the class of an object to be serialized
- **IOException** -- if an I/O error occurs during serialization

Example

The following example shows the usage of **java.io.ObjectOutputStream.writeUnshared** method.

```
package com.tutorialspoint;  
  
import java.io.*;  
  
public class ObjectOutputStreamDemo {
```

```
public static void main(String[] args) {  
  
    Object s = "Hello World!";  
    Object i = 1974;  
    try {  
  
        // create a new file with an ObjectOutputStream  
        FileOutputStream out = new FileOutputStream("test.txt");  
        ObjectOutputStream oout = new ObjectOutputStream(out);  
  
        // write something in the file  
        oout.writeUnshared(s);  
        oout.writeUnshared(i);  
  
        // close the stream  
        oout.close();  
  
        // create an ObjectInputStream for the file we created before  
        ObjectInputStream ois =  
            new ObjectInputStream(new FileInputStream("test.txt"));  
  
        // read and print what we wrote before  
        System.out.println("" + ois.readUnshared());  
        System.out.println("" + ois.readUnshared());  
  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}  
}
```

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

```
Hello World!  
1974
```

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