

# JAVA.IO.OBJECTINPUTSTREAM.READUNSHARED METHOD

[http://www.tutorialspoint.com/java/io/objectinputstream\\_readunshared.htm](http://www.tutorialspoint.com/java/io/objectinputstream_readunshared.htm)

Copyright © tutorialspoint.com

## Description

The **java.io.ObjectInputStream.readUnshared** method reads an "unshared" object from the ObjectInputStream. This method is identical to readObject, except that it prevents subsequent calls to readObject and readUnshared from returning additional references to the deserialized instance obtained via this call. Specifically:

- If readUnshared is called to deserialize a back-reference *thestreamrepresentationofanobjectwhichhasbeenwrittenpreviouslytothestream*, an ObjectStreamException will be thrown
- If readUnshared returns successfully, then any subsequent attempts to deserialize back-references to the stream handle deserialized by readUnshared will cause an ObjectStreamException to be thrown.

Deserializing an object via readUnshared invalidates the stream handle associated with the returned object. Note that this in itself does not always guarantee that the reference returned by readUnshared is unique; the deserialized object may define a readResolve method which returns an object visible to other parties, or readUnshared may return a Class object or enum constant obtainable elsewhere in the stream or through external means. If the deserialized object defines a readResolve method and the invocation of that method returns an array, then readUnshared returns a shallow clone of that array; this guarantees that the returned array object is unique and cannot be obtained a second time from an invocation of readObject or readUnshared on the ObjectInputStream, even if the underlying data stream has been manipulated.

ObjectInputStream 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.ObjectInputStream.readUnshared** method

```
public Object readUnshared()
```

## Parameters

- NA

## Return Value

This method returns a reference to deserialized object

## Exception

- **ClassNotFoundException** -- if class of an object to deserialize cannot be found
- **StreamCorruptedException** -- if control information in the stream is inconsistent
- **ObjectStreamException** -- if object to deserialize has already appeared in stream
- **OptionalDataException** -- if primitive data is next in stream
- **IOException** -- if an I/O error occurs during deserialization

## Example

The following example shows the usage of **java.io.ObjectInputStream.readUnshared** method.

```
package com.tutorialspoint;
```

```
import java.io.*;

public class ObjectInputStreamDemo {

    public static void main(String[] args) {

        String s = "Hello World";
        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.flush();

            // create an ObjectInputStream for the file we created before
            ObjectInputStream ois =
                new ObjectInputStream(new FileInputStream("test.txt"));

            // read and print the unshared object
            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

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