| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/OutputStream.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/io/OptionalDataException.html)   [**NEXT CLASS**](http://docs.google.com/java/io/OutputStreamWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/OutputStream.html)    [**NO FRAMES**](http://docs.google.com/OutputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.io**

Class OutputStream

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.io.OutputStream**

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Flushable](http://docs.google.com/java/io/Flushable.html) **Direct Known Subclasses:** [ByteArrayOutputStream](http://docs.google.com/java/io/ByteArrayOutputStream.html), [FileOutputStream](http://docs.google.com/java/io/FileOutputStream.html), [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html), [ObjectOutputStream](http://docs.google.com/java/io/ObjectOutputStream.html), [OutputStream](http://docs.google.com/org/omg/CORBA/portable/OutputStream.html), [PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html)

public abstract class **OutputStream**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Closeable](http://docs.google.com/java/io/Closeable.html), [Flushable](http://docs.google.com/java/io/Flushable.html)

This abstract class is the superclass of all classes representing an output stream of bytes. An output stream accepts output bytes and sends them to some sink.

Applications that need to define a subclass of OutputStream must always provide at least a method that writes one byte of output.

**Since:** JDK1.0 **See Also:**[BufferedOutputStream](http://docs.google.com/java/io/BufferedOutputStream.html), [ByteArrayOutputStream](http://docs.google.com/java/io/ByteArrayOutputStream.html), [DataOutputStream](http://docs.google.com/java/io/DataOutputStream.html), [FilterOutputStream](http://docs.google.com/java/io/FilterOutputStream.html), [InputStream](http://docs.google.com/java/io/InputStream.html), [write(int)](http://docs.google.com/java/io/OutputStream.html#write(int))

| **Constructor Summary** | |
| --- | --- |
| [**OutputStream**](http://docs.google.com/java/io/OutputStream.html#OutputStream())() |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/io/OutputStream.html#close())()            Closes this output stream and releases any system resources associated with this stream. |
| void | [**flush**](http://docs.google.com/java/io/OutputStream.html#flush())()            Flushes this output stream and forces any buffered output bytes to be written out. |
| void | [**write**](http://docs.google.com/java/io/OutputStream.html#write(byte%5B%5D))(byte[] b)            Writes b.length bytes from the specified byte array to this output stream. |
| void | [**write**](http://docs.google.com/java/io/OutputStream.html#write(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Writes len bytes from the specified byte array starting at offset off to this output stream. |
| abstract  void | [**write**](http://docs.google.com/java/io/OutputStream.html#write(int))(int b)            Writes the specified byte to this output stream. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### OutputStream

public **OutputStream**()

| **Method Detail** |
| --- |

### write

public abstract void **write**(int b)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes the specified byte to this output stream. The general contract for write is that one byte is written to the output stream. The byte to be written is the eight low-order bits of the argument b. The 24 high-order bits of b are ignored.

Subclasses of OutputStream must provide an implementation for this method.

**Parameters:**b - the byte. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs. In particular, an IOException may be thrown if the output stream has been closed.

### write

public void **write**(byte[] b)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes b.length bytes from the specified byte array to this output stream. The general contract for write(b) is that it should have exactly the same effect as the call write(b, 0, b.length).

**Parameters:**b - the data. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**See Also:**[write(byte[], int, int)](http://docs.google.com/java/io/OutputStream.html#write(byte%5B%5D,%20int,%20int))

### write

public void **write**(byte[] b,  
 int off,  
 int len)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes len bytes from the specified byte array starting at offset off to this output stream. The general contract for write(b, off, len) is that some of the bytes in the array b are written to the output stream in order; element b[off] is the first byte written and b[off+len-1] is the last byte written by this operation.

The write method of OutputStream calls the write method of one argument on each of the bytes to be written out. Subclasses are encouraged to override this method and provide a more efficient implementation.

If b is null, a NullPointerException is thrown.

If off is negative, or len is negative, or off+len is greater than the length of the array b, then an IndexOutOfBoundsException is thrown.

**Parameters:**b - the data.off - the start offset in the data.len - the number of bytes to write. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs. In particular, an IOException is thrown if the output stream is closed.

### flush

public void **flush**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Flushes this output stream and forces any buffered output bytes to be written out. The general contract of flush is that calling it is an indication that, if any bytes previously written have been buffered by the implementation of the output stream, such bytes should immediately be written to their intended destination.

If the intended destination of this stream is an abstraction provided by the underlying operating system, for example a file, then flushing the stream guarantees only that bytes previously written to the stream are passed to the operating system for writing; it does not guarantee that they are actually written to a physical device such as a disk drive.

The flush method of OutputStream does nothing.

**Specified by:**[flush](http://docs.google.com/java/io/Flushable.html#flush()) in interface [Flushable](http://docs.google.com/java/io/Flushable.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### close

public void **close**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Closes this output stream and releases any system resources associated with this stream. The general contract of close is that it closes the output stream. A closed stream cannot perform output operations and cannot be reopened.

The close method of OutputStream does nothing.

**Specified by:**[close](http://docs.google.com/java/io/Closeable.html#close()) in interface [Closeable](http://docs.google.com/java/io/Closeable.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/OutputStream.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/io/OptionalDataException.html)   [**NEXT CLASS**](http://docs.google.com/java/io/OutputStreamWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/OutputStream.html)    [**NO FRAMES**](http://docs.google.com/OutputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).