| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/FileOutputStream.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/FileNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/java/io/FilePermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/FileOutputStream.html)    [**NO FRAMES**](http://docs.google.com/FileOutputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#lnxbz9) |

## **java.io**

Class FileOutputStream

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

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Flushable](http://docs.google.com/java/io/Flushable.html)

public class **FileOutputStream**extends [OutputStream](http://docs.google.com/java/io/OutputStream.html)

A file output stream is an output stream for writing data to a File or to a FileDescriptor. Whether or not a file is available or may be created depends upon the underlying platform. Some platforms, in particular, allow a file to be opened for writing by only one FileOutputStream (or other file-writing object) at a time. In such situations the constructors in this class will fail if the file involved is already open.

FileOutputStream is meant for writing streams of raw bytes such as image data. For writing streams of characters, consider using FileWriter.

**Since:** JDK1.0 **See Also:**[File](http://docs.google.com/java/io/File.html), [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html), [FileInputStream](http://docs.google.com/java/io/FileInputStream.html)

| **Constructor Summary** | |
| --- | --- |
| [**FileOutputStream**](http://docs.google.com/java/io/FileOutputStream.html#FileOutputStream(java.io.File))([File](http://docs.google.com/java/io/File.html) file)            Creates a file output stream to write to the file represented by the specified File object. |
| [**FileOutputStream**](http://docs.google.com/java/io/FileOutputStream.html#FileOutputStream(java.io.File,%20boolean))([File](http://docs.google.com/java/io/File.html) file, boolean append)            Creates a file output stream to write to the file represented by the specified File object. |
| [**FileOutputStream**](http://docs.google.com/java/io/FileOutputStream.html#FileOutputStream(java.io.FileDescriptor))([FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) fdObj)            Creates an output file stream to write to the specified file descriptor, which represents an existing connection to an actual file in the file system. |
| [**FileOutputStream**](http://docs.google.com/java/io/FileOutputStream.html#FileOutputStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Creates an output file stream to write to the file with the specified name. |
| [**FileOutputStream**](http://docs.google.com/java/io/FileOutputStream.html#FileOutputStream(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) name, boolean append)            Creates an output file stream to write to the file with the specified name. |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/io/FileOutputStream.html#close())()            Closes this file output stream and releases any system resources associated with this stream. |
| protected  void | [**finalize**](http://docs.google.com/java/io/FileOutputStream.html#finalize())()            Cleans up the connection to the file, and ensures that the close method of this file output stream is called when there are no more references to this stream. |
| [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) | [**getChannel**](http://docs.google.com/java/io/FileOutputStream.html#getChannel())()            Returns the unique [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) object associated with this file output stream. |
| [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) | [**getFD**](http://docs.google.com/java/io/FileOutputStream.html#getFD())()            Returns the file descriptor associated with this stream. |
| void | [**write**](http://docs.google.com/java/io/FileOutputStream.html#write(byte%5B%5D))(byte[] b)            Writes b.length bytes from the specified byte array to this file output stream. |
| void | [**write**](http://docs.google.com/java/io/FileOutputStream.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 file output stream. |
| void | [**write**](http://docs.google.com/java/io/FileOutputStream.html#write(int))(int b)            Writes the specified byte to this file output stream. |

| **Methods inherited from class java.io.**[**OutputStream**](http://docs.google.com/java/io/OutputStream.html) |
| --- |
| [flush](http://docs.google.com/java/io/OutputStream.html#flush()) |

| **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)), [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** |
| --- |

### FileOutputStream

public **FileOutputStream**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates an output file stream to write to the file with the specified name. A new FileDescriptor object is created to represent this file connection.

First, if there is a security manager, its checkWrite method is called with name as its argument.

If the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason then a FileNotFoundException is thrown.

**Parameters:**name - the system-dependent filename **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - if the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkWrite method denies write access to the file.**See Also:**[SecurityManager.checkWrite(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.lang.String))

### FileOutputStream

public **FileOutputStream**([String](http://docs.google.com/java/lang/String.html) name,  
 boolean append)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates an output file stream to write to the file with the specified name. If the second argument is true, then bytes will be written to the end of the file rather than the beginning. A new FileDescriptor object is created to represent this file connection.

First, if there is a security manager, its checkWrite method is called with name as its argument.

If the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason then a FileNotFoundException is thrown.

**Parameters:**name - the system-dependent file nameappend - if true, then bytes will be written to the end of the file rather than the beginning **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - if the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkWrite method denies write access to the file.**Since:** JDK1.1 **See Also:**[SecurityManager.checkWrite(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.lang.String))

### FileOutputStream

public **FileOutputStream**([File](http://docs.google.com/java/io/File.html) file)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates a file output stream to write to the file represented by the specified File object. A new FileDescriptor object is created to represent this file connection.

First, if there is a security manager, its checkWrite method is called with the path represented by the file argument as its argument.

If the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason then a FileNotFoundException is thrown.

**Parameters:**file - the file to be opened for writing. **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - if the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkWrite method denies write access to the file.**See Also:**[File.getPath()](http://docs.google.com/java/io/File.html#getPath()), [SecurityException](http://docs.google.com/java/lang/SecurityException.html), [SecurityManager.checkWrite(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.lang.String))

### FileOutputStream

public **FileOutputStream**([File](http://docs.google.com/java/io/File.html) file,  
 boolean append)  
 throws [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html)

Creates a file output stream to write to the file represented by the specified File object. If the second argument is true, then bytes will be written to the end of the file rather than the beginning. A new FileDescriptor object is created to represent this file connection.

First, if there is a security manager, its checkWrite method is called with the path represented by the file argument as its argument.

If the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason then a FileNotFoundException is thrown.

**Parameters:**file - the file to be opened for writing.append - if true, then bytes will be written to the end of the file rather than the beginning **Throws:** [FileNotFoundException](http://docs.google.com/java/io/FileNotFoundException.html) - if the file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkWrite method denies write access to the file.**Since:** 1.4 **See Also:**[File.getPath()](http://docs.google.com/java/io/File.html#getPath()), [SecurityException](http://docs.google.com/java/lang/SecurityException.html), [SecurityManager.checkWrite(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.lang.String))

### FileOutputStream

public **FileOutputStream**([FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) fdObj)

Creates an output file stream to write to the specified file descriptor, which represents an existing connection to an actual file in the file system.

First, if there is a security manager, its checkWrite method is called with the file descriptor fdObj argument as its argument.

**Parameters:**fdObj - the file descriptor to be opened for writing **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkWrite method denies write access to the file descriptor**See Also:**[SecurityManager.checkWrite(java.io.FileDescriptor)](http://docs.google.com/java/lang/SecurityManager.html#checkWrite(java.io.FileDescriptor))

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

### write

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

Writes the specified byte to this file output stream. Implements the write method of OutputStream.

**Specified by:**[write](http://docs.google.com/java/io/OutputStream.html#write(int)) in class [OutputStream](http://docs.google.com/java/io/OutputStream.html) **Parameters:**b - the byte to be written. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### 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 file output stream.

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

**Overrides:**[write](http://docs.google.com/java/io/OutputStream.html#write(byte%5B%5D,%20int,%20int)) in class [OutputStream](http://docs.google.com/java/io/OutputStream.html) **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.

### close

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

Closes this file output stream and releases any system resources associated with this stream. This file output stream may no longer be used for writing bytes.

If this stream has an associated channel then the channel is closed as well.

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

### getFD

public final [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) **getFD**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns the file descriptor associated with this stream.

**Returns:**the FileDescriptor object that represents the connection to the file in the file system being used by this FileOutputStream object. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**See Also:**[FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html)

### getChannel

public [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) **getChannel**()

Returns the unique [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) object associated with this file output stream.

The initial [position](http://docs.google.com/java/nio/channels/FileChannel.html#position()) of the returned channel will be equal to the number of bytes written to the file so far unless this stream is in append mode, in which case it will be equal to the size of the file. Writing bytes to this stream will increment the channel's position accordingly. Changing the channel's position, either explicitly or by writing, will change this stream's file position.

**Returns:**the file channel associated with this file output stream**Since:** 1.4

### finalize

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

Cleans up the connection to the file, and ensures that the close method of this file output stream is called when there are no more references to this stream.

**Overrides:**[finalize](http://docs.google.com/java/lang/Object.html#finalize()) in class [Object](http://docs.google.com/java/lang/Object.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**See Also:**[FileInputStream.close()](http://docs.google.com/java/io/FileInputStream.html#close())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/FileOutputStream.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/FileNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/java/io/FilePermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/FileOutputStream.html)    [**NO FRAMES**](http://docs.google.com/FileOutputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#lnxbz9) |

[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).