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

## **java.io**

Class FileInputStream

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

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

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

A FileInputStream obtains input bytes from a file in a file system. What files are available depends on the host environment.

FileInputStream is meant for reading streams of raw bytes such as image data. For reading streams of characters, consider using FileReader.

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

| **Constructor Summary** | |
| --- | --- |
| [**FileInputStream**](http://docs.google.com/java/io/FileInputStream.html#FileInputStream(java.io.File))([File](http://docs.google.com/java/io/File.html) file)            Creates a FileInputStream by opening a connection to an actual file, the file named by the File object file in the file system. |
| [**FileInputStream**](http://docs.google.com/java/io/FileInputStream.html#FileInputStream(java.io.FileDescriptor))([FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) fdObj)            Creates a FileInputStream by using the file descriptor fdObj, which represents an existing connection to an actual file in the file system. |
| [**FileInputStream**](http://docs.google.com/java/io/FileInputStream.html#FileInputStream(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Creates a FileInputStream by opening a connection to an actual file, the file named by the path name name in the file system. |

| **Method Summary** | |
| --- | --- |
| int | [**available**](http://docs.google.com/java/io/FileInputStream.html#available())()            Returns an estimate of the number of remaining bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream. |
| void | [**close**](http://docs.google.com/java/io/FileInputStream.html#close())()            Closes this file input stream and releases any system resources associated with the stream. |
| protected  void | [**finalize**](http://docs.google.com/java/io/FileInputStream.html#finalize())()            Ensures that the close method of this file input stream is called when there are no more references to it. |
| [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) | [**getChannel**](http://docs.google.com/java/io/FileInputStream.html#getChannel())()            Returns the unique [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) object associated with this file input stream. |
| [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) | [**getFD**](http://docs.google.com/java/io/FileInputStream.html#getFD())()            Returns the FileDescriptor object that represents the connection to the actual file in the file system being used by this FileInputStream. |
| int | [**read**](http://docs.google.com/java/io/FileInputStream.html#read())()            Reads a byte of data from this input stream. |
| int | [**read**](http://docs.google.com/java/io/FileInputStream.html#read(byte%5B%5D))(byte[] b)            Reads up to b.length bytes of data from this input stream into an array of bytes. |
| int | [**read**](http://docs.google.com/java/io/FileInputStream.html#read(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Reads up to len bytes of data from this input stream into an array of bytes. |
| long | [**skip**](http://docs.google.com/java/io/FileInputStream.html#skip(long))(long n)            Skips over and discards n bytes of data from the input stream. |

| **Methods inherited from class java.io.**[**InputStream**](http://docs.google.com/java/io/InputStream.html) |
| --- |
| [mark](http://docs.google.com/java/io/InputStream.html#mark(int)), [markSupported](http://docs.google.com/java/io/InputStream.html#markSupported()), [reset](http://docs.google.com/java/io/InputStream.html#reset()) |

| **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** |
| --- |

### FileInputStream

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

Creates a FileInputStream by opening a connection to an actual file, the file named by the path name name in the file system. A new FileDescriptor object is created to represent this file connection.

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

If the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading then a FileNotFoundException is thrown.

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

### FileInputStream

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

Creates a FileInputStream by opening a connection to an actual file, the file named by the File object file in the file system. A new FileDescriptor object is created to represent this file connection.

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

If the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading then a FileNotFoundException is thrown.

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

### FileInputStream

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

Creates a FileInputStream by using the file descriptor fdObj, which represents an existing connection to an actual file in the file system.

If there is a security manager, its checkRead method is called with the file descriptor fdObj as its argument to see if it's ok to read the file descriptor. If read access is denied to the file descriptor a SecurityException is thrown.

If fdObj is null then a NullPointerException is thrown.

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

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

### read

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

Reads a byte of data from this input stream. This method blocks if no input is yet available.

**Specified by:**[read](http://docs.google.com/java/io/InputStream.html#read()) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Returns:**the next byte of data, or -1 if the end of the file is reached. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### read

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

Reads up to b.length bytes of data from this input stream into an array of bytes. This method blocks until some input is available.

**Overrides:**[read](http://docs.google.com/java/io/InputStream.html#read(byte%5B%5D)) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Parameters:**b - the buffer into which the data is read. **Returns:**the total number of bytes read into the buffer, or -1 if there is no more data because the end of the file has been reached. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**See Also:**[InputStream.read(byte[], int, int)](http://docs.google.com/java/io/InputStream.html#read(byte%5B%5D,%20int,%20int))

### read

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

Reads up to len bytes of data from this input stream into an array of bytes. If len is not zero, the method blocks until some input is available; otherwise, no bytes are read and 0 is returned.

**Overrides:**[read](http://docs.google.com/java/io/InputStream.html#read(byte%5B%5D,%20int,%20int)) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Parameters:**b - the buffer into which the data is read.off - the start offset in the destination array blen - the maximum number of bytes read. **Returns:**the total number of bytes read into the buffer, or -1 if there is no more data because the end of the file has been reached. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If b is null. [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If off is negative, len is negative, or len is greater than b.length - off [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**See Also:**[InputStream.read()](http://docs.google.com/java/io/InputStream.html#read())

### skip

public long **skip**(long n)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Skips over and discards n bytes of data from the input stream.

The skip method may, for a variety of reasons, end up skipping over some smaller number of bytes, possibly 0. If n is negative, an IOException is thrown, even though the skip method of the [InputStream](http://docs.google.com/java/io/InputStream.html) superclass does nothing in this case. The actual number of bytes skipped is returned.

This method may skip more bytes than are remaining in the backing file. This produces no exception and the number of bytes skipped may include some number of bytes that were beyond the EOF of the backing file. Attempting to read from the stream after skipping past the end will result in -1 indicating the end of the file.

**Overrides:**[skip](http://docs.google.com/java/io/InputStream.html#skip(long)) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Parameters:**n - the number of bytes to be skipped. **Returns:**the actual number of bytes skipped. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if n is negative, if the stream does not support seek, or if an I/O error occurs.

### available

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

Returns an estimate of the number of remaining bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream. The next invocation might be the same thread or another thread. A single read or skip of this many bytes will not block, but may read or skip fewer bytes.

In some cases, a non-blocking read (or skip) may appear to be blocked when it is merely slow, for example when reading large files over slow networks.

**Overrides:**[available](http://docs.google.com/java/io/InputStream.html#available()) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Returns:**an estimate of the number of remaining bytes that can be read (or skipped over) from this input stream without blocking. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if this file input stream has been closed by calling close or an I/O error occurs.

### close

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

Closes this file input stream and releases any system resources associated with the stream.

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/InputStream.html#close()) in class [InputStream](http://docs.google.com/java/io/InputStream.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 FileDescriptor object that represents the connection to the actual file in the file system being used by this FileInputStream.

**Returns:**the file descriptor object associated with this stream. **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 input 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 read from the file so far. Reading bytes from this stream will increment the channel's position. Changing the channel's position, either explicitly or by reading, will change this stream's file position.

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

### finalize

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

Ensures that the close method of this file input stream is called when there are no more references to it.

**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:**[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/FileInputStream.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/FileFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/FilenameFilter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/FileInputStream.html)    [**NO FRAMES**](http://docs.google.com/FileInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#3rdcrjn) |

[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).