| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SequenceInputStream.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/Reader.html)   [**NEXT CLASS**](http://docs.google.com/java/io/Serializable.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/SequenceInputStream.html)    [**NO FRAMES**](http://docs.google.com/SequenceInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.io**

Class SequenceInputStream

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

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

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

A SequenceInputStream represents the logical concatenation of other input streams. It starts out with an ordered collection of input streams and reads from the first one until end of file is reached, whereupon it reads from the second one, and so on, until end of file is reached on the last of the contained input streams.

**Since:** JDK1.0

| **Constructor Summary** | |
| --- | --- |
| [**SequenceInputStream**](http://docs.google.com/java/io/SequenceInputStream.html#SequenceInputStream(java.util.Enumeration))([Enumeration](http://docs.google.com/java/util/Enumeration.html)<? extends [InputStream](http://docs.google.com/java/io/InputStream.html)> e)            Initializes a newly created SequenceInputStream by remembering the argument, which must be an Enumeration that produces objects whose run-time type is InputStream. |
| [**SequenceInputStream**](http://docs.google.com/java/io/SequenceInputStream.html#SequenceInputStream(java.io.InputStream,%20java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) s1, [InputStream](http://docs.google.com/java/io/InputStream.html) s2)            Initializes a newly created SequenceInputStream by remembering the two arguments, which will be read in order, first s1 and then s2, to provide the bytes to be read from this SequenceInputStream. |

| **Method Summary** | |
| --- | --- |
| int | [**available**](http://docs.google.com/java/io/SequenceInputStream.html#available())()            Returns an estimate of the number of bytes that can be read (or skipped over) from the current underlying input stream without blocking by the next invocation of a method for the current underlying input stream. |
| void | [**close**](http://docs.google.com/java/io/SequenceInputStream.html#close())()            Closes this input stream and releases any system resources associated with the stream. |
| int | [**read**](http://docs.google.com/java/io/SequenceInputStream.html#read())()            Reads the next byte of data from this input stream. |
| int | [**read**](http://docs.google.com/java/io/SequenceInputStream.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. |

| **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()), [read](http://docs.google.com/java/io/InputStream.html#read(byte%5B%5D)), [reset](http://docs.google.com/java/io/InputStream.html#reset()), [skip](http://docs.google.com/java/io/InputStream.html#skip(long)) |

| **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** |
| --- |

### SequenceInputStream

public **SequenceInputStream**([Enumeration](http://docs.google.com/java/util/Enumeration.html)<? extends [InputStream](http://docs.google.com/java/io/InputStream.html)> e)

Initializes a newly created SequenceInputStream by remembering the argument, which must be an Enumeration that produces objects whose run-time type is InputStream. The input streams that are produced by the enumeration will be read, in order, to provide the bytes to be read from this SequenceInputStream. After each input stream from the enumeration is exhausted, it is closed by calling its close method.

**Parameters:**e - an enumeration of input streams.**See Also:**[Enumeration](http://docs.google.com/java/util/Enumeration.html)

### SequenceInputStream

public **SequenceInputStream**([InputStream](http://docs.google.com/java/io/InputStream.html) s1,  
 [InputStream](http://docs.google.com/java/io/InputStream.html) s2)

Initializes a newly created SequenceInputStream by remembering the two arguments, which will be read in order, first s1 and then s2, to provide the bytes to be read from this SequenceInputStream.

**Parameters:**s1 - the first input stream to read.s2 - the second input stream to read.

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

### available

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

Returns an estimate of the number of bytes that can be read (or skipped over) from the current underlying input stream without blocking by the next invocation of a method for the current underlying 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.

This method simply calls available of the current underlying input stream and returns the result.

**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 bytes that can be read (or skipped over) from the current underlying input stream without blocking or 0 if this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/SequenceInputStream.html#close()) method **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**Since:** JDK1.1

### read

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

Reads the next byte of data from this input stream. The byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.

This method tries to read one character from the current substream. If it reaches the end of the stream, it calls the close method of the current substream and begins reading from the next substream.

**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 stream is reached. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### 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 at least 1 byte of input is available; otherwise, no bytes are read and 0 is returned.

The read method of SequenceInputStream tries to read the data from the current substream. If it fails to read any characters because the substream has reached the end of the stream, it calls the close method of the current substream and begins reading from the next substream.

**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 array b at which the data is written.len - the maximum number of bytes read. **Returns:**int the number of bytes read. **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())

### close

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

Closes this input stream and releases any system resources associated with the stream. A closed SequenceInputStream cannot perform input operations and cannot be reopened.

If this stream was created from an enumeration, all remaining elements are requested from the enumeration and closed before the close method returns.

**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.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SequenceInputStream.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/Reader.html)   [**NEXT CLASS**](http://docs.google.com/java/io/Serializable.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/SequenceInputStream.html)    [**NO FRAMES**](http://docs.google.com/SequenceInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

[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).