| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PipedInputStream.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/OutputStreamWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PipedOutputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PipedInputStream.html)    [**NO FRAMES**](http://docs.google.com/PipedInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#3j2qqm3) |

## **java.io**

Class PipedInputStream

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

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

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

A piped input stream should be connected to a piped output stream; the piped input stream then provides whatever data bytes are written to the piped output stream. Typically, data is read from a PipedInputStream object by one thread and data is written to the corresponding PipedOutputStream by some other thread. Attempting to use both objects from a single thread is not recommended, as it may deadlock the thread. The piped input stream contains a buffer, decoupling read operations from write operations, within limits. A pipe is said to be  *broken* if a thread that was providing data bytes to the connected piped output stream is no longer alive.

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

| **Field Summary** | |
| --- | --- |
| protected  byte[] | [**buffer**](http://docs.google.com/java/io/PipedInputStream.html#buffer)            The circular buffer into which incoming data is placed. |
| protected  int | [**in**](http://docs.google.com/java/io/PipedInputStream.html#in)            The index of the position in the circular buffer at which the next byte of data will be stored when received from the connected piped output stream. |
| protected  int | [**out**](http://docs.google.com/java/io/PipedInputStream.html#out)            The index of the position in the circular buffer at which the next byte of data will be read by this piped input stream. |
| protected static int | [**PIPE\_SIZE**](http://docs.google.com/java/io/PipedInputStream.html#PIPE_SIZE)            The default size of the pipe's circular input buffer. |

| **Constructor Summary** | |
| --- | --- |
| [**PipedInputStream**](http://docs.google.com/java/io/PipedInputStream.html#PipedInputStream())()            Creates a PipedInputStream so that it is not yet [connected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)). |
| [**PipedInputStream**](http://docs.google.com/java/io/PipedInputStream.html#PipedInputStream(int))(int pipeSize)            Creates a PipedInputStream so that it is not yet [connected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)) and uses the specified pipe size for the pipe's buffer. |
| [**PipedInputStream**](http://docs.google.com/java/io/PipedInputStream.html#PipedInputStream(java.io.PipedOutputStream))([PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html) src)            Creates a PipedInputStream so that it is connected to the piped output stream src. |
| [**PipedInputStream**](http://docs.google.com/java/io/PipedInputStream.html#PipedInputStream(java.io.PipedOutputStream,%20int))([PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html) src, int pipeSize)            Creates a PipedInputStream so that it is connected to the piped output stream src and uses the specified pipe size for the pipe's buffer. |

| **Method Summary** | |
| --- | --- |
| int | [**available**](http://docs.google.com/java/io/PipedInputStream.html#available())()            Returns the number of bytes that can be read from this input stream without blocking. |
| void | [**close**](http://docs.google.com/java/io/PipedInputStream.html#close())()            Closes this piped input stream and releases any system resources associated with the stream. |
| void | [**connect**](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream))([PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html) src)            Causes this piped input stream to be connected to the piped output stream src. |
| int | [**read**](http://docs.google.com/java/io/PipedInputStream.html#read())()            Reads the next byte of data from this piped input stream. |
| int | [**read**](http://docs.google.com/java/io/PipedInputStream.html#read(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Reads up to len bytes of data from this piped input stream into an array of bytes. |
| protected  void | [**receive**](http://docs.google.com/java/io/PipedInputStream.html#receive(int))(int b)            Receives a byte of data. |

| **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)) |

| **Field Detail** |
| --- |

### PIPE\_SIZE

protected static final int **PIPE\_SIZE**

The default size of the pipe's circular input buffer.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.io.PipedInputStream.PIPE_SIZE)

### buffer

protected byte[] **buffer**

The circular buffer into which incoming data is placed.

**Since:** JDK1.1

### in

protected int **in**

The index of the position in the circular buffer at which the next byte of data will be stored when received from the connected piped output stream. in<0 implies the buffer is empty, in==out implies the buffer is full

**Since:** JDK1.1

### out

protected int **out**

The index of the position in the circular buffer at which the next byte of data will be read by this piped input stream.

**Since:** JDK1.1

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

### PipedInputStream

public **PipedInputStream**([PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html) src)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates a PipedInputStream so that it is connected to the piped output stream src. Data bytes written to src will then be available as input from this stream.

**Parameters:**src - the stream to connect to. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### PipedInputStream

public **PipedInputStream**([PipedOutputStream](http://docs.google.com/java/io/PipedOutputStream.html) src,  
 int pipeSize)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates a PipedInputStream so that it is connected to the piped output stream src and uses the specified pipe size for the pipe's buffer. Data bytes written to src will then be available as input from this stream.

**Parameters:**src - the stream to connect to.pipeSize - the size of the pipe's buffer. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if pipeSize <= 0.**Since:** 1.6

### PipedInputStream

public **PipedInputStream**()

Creates a PipedInputStream so that it is not yet [connected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)). It must be [connected](http://docs.google.com/java/io/PipedOutputStream.html#connect(java.io.PipedInputStream)) to a PipedOutputStream before being used.

### PipedInputStream

public **PipedInputStream**(int pipeSize)

Creates a PipedInputStream so that it is not yet [connected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)) and uses the specified pipe size for the pipe's buffer. It must be [connected](http://docs.google.com/java/io/PipedOutputStream.html#connect(java.io.PipedInputStream)) to a PipedOutputStream before being used.

**Parameters:**pipeSize - the size of the pipe's buffer. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if pipeSize <= 0.**Since:** 1.6

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

### connect

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

Causes this piped input stream to be connected to the piped output stream src. If this object is already connected to some other piped output stream, an IOException is thrown.

If src is an unconnected piped output stream and snk is an unconnected piped input stream, they may be connected by either the call:

snk.connect(src)

or the call:

src.connect(snk)

The two calls have the same effect.

**Parameters:**src - The piped output stream to connect to. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### receive

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

Receives a byte of data. This method will block if no input is available.

**Parameters:**b - the byte being received **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If the pipe is  [broken](#3znysh7), [unconnected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)), closed, or 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 piped input stream. The value byte is returned as an int in the range 0 to 255. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.

**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 the pipe is [unconnected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)),  [broken](#3znysh7), closed, or 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 piped input stream into an array of bytes. Less than len bytes will be read if the end of the data stream is reached or if len exceeds the pipe's buffer size. If len is zero, then no bytes are read and 0 is returned; otherwise, the method blocks until at least 1 byte of input is available, end of the stream has been detected, or an exception is thrown.

**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 stream 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 the pipe is  [broken](#3znysh7), [unconnected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)), closed, or if an I/O error occurs.**See Also:**[InputStream.read()](http://docs.google.com/java/io/InputStream.html#read())

### available

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

Returns the number of bytes that can be read from this input stream without blocking.

**Overrides:**[available](http://docs.google.com/java/io/InputStream.html#available()) in class [InputStream](http://docs.google.com/java/io/InputStream.html) **Returns:**the number of bytes that can be read from this input stream without blocking, or 0 if this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PipedInputStream.html#close()) method, or if the pipe is [unconnected](http://docs.google.com/java/io/PipedInputStream.html#connect(java.io.PipedOutputStream)), or  [broken](#3znysh7). **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.**Since:** JDK1.0.2

### close

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

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

**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/PipedInputStream.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/OutputStreamWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PipedOutputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PipedInputStream.html)    [**NO FRAMES**](http://docs.google.com/PipedInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#3j2qqm3) |

[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).