| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PushbackInputStream.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/PrintWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PushbackReader.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PushbackInputStream.html)    [**NO FRAMES**](http://docs.google.com/PushbackInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

## **java.io**

Class PushbackInputStream

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

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

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

A PushbackInputStream adds functionality to another input stream, namely the ability to "push back" or "unread" one byte. This is useful in situations where it is convenient for a fragment of code to read an indefinite number of data bytes that are delimited by a particular byte value; after reading the terminating byte, the code fragment can "unread" it, so that the next read operation on the input stream will reread the byte that was pushed back. For example, bytes representing the characters constituting an identifier might be terminated by a byte representing an operator character; a method whose job is to read just an identifier can read until it sees the operator and then push the operator back to be re-read.

**Since:** JDK1.0

| **Field Summary** | |
| --- | --- |
| protected  byte[] | [**buf**](http://docs.google.com/java/io/PushbackInputStream.html#buf)            The pushback buffer. |
| protected  int | [**pos**](http://docs.google.com/java/io/PushbackInputStream.html#pos)            The position within the pushback buffer from which the next byte will be read. |

| **Fields inherited from class java.io.**[**FilterInputStream**](http://docs.google.com/java/io/FilterInputStream.html) |
| --- |
| [in](http://docs.google.com/java/io/FilterInputStream.html#in) |

| **Constructor Summary** | |
| --- | --- |
| [**PushbackInputStream**](http://docs.google.com/java/io/PushbackInputStream.html#PushbackInputStream(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) in)            Creates a PushbackInputStream and saves its argument, the input stream in, for later use. |
| [**PushbackInputStream**](http://docs.google.com/java/io/PushbackInputStream.html#PushbackInputStream(java.io.InputStream,%20int))([InputStream](http://docs.google.com/java/io/InputStream.html) in, int size)            Creates a PushbackInputStream with a pushback buffer of the specified size, and saves its argument, the input stream in, for later use. |

| **Method Summary** | |
| --- | --- |
| int | [**available**](http://docs.google.com/java/io/PushbackInputStream.html#available())()            Returns an estimate of the number of 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/PushbackInputStream.html#close())()            Closes this input stream and releases any system resources associated with the stream. |
| void | [**mark**](http://docs.google.com/java/io/PushbackInputStream.html#mark(int))(int readlimit)            Marks the current position in this input stream. |
| boolean | [**markSupported**](http://docs.google.com/java/io/PushbackInputStream.html#markSupported())()            Tests if this input stream supports the mark and reset methods, which it does not. |
| int | [**read**](http://docs.google.com/java/io/PushbackInputStream.html#read())()            Reads the next byte of data from this input stream. |
| int | [**read**](http://docs.google.com/java/io/PushbackInputStream.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. |
| void | [**reset**](http://docs.google.com/java/io/PushbackInputStream.html#reset())()            Repositions this stream to the position at the time the mark method was last called on this input stream. |
| long | [**skip**](http://docs.google.com/java/io/PushbackInputStream.html#skip(long))(long n)            Skips over and discards n bytes of data from this input stream. |
| void | [**unread**](http://docs.google.com/java/io/PushbackInputStream.html#unread(byte%5B%5D))(byte[] b)            Pushes back an array of bytes by copying it to the front of the pushback buffer. |
| void | [**unread**](http://docs.google.com/java/io/PushbackInputStream.html#unread(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Pushes back a portion of an array of bytes by copying it to the front of the pushback buffer. |
| void | [**unread**](http://docs.google.com/java/io/PushbackInputStream.html#unread(int))(int b)            Pushes back a byte by copying it to the front of the pushback buffer. |

| **Methods inherited from class java.io.**[**FilterInputStream**](http://docs.google.com/java/io/FilterInputStream.html) |
| --- |
| [read](http://docs.google.com/java/io/FilterInputStream.html#read(byte%5B%5D)) |

| **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** |
| --- |

### buf

protected byte[] **buf**

The pushback buffer.

**Since:** JDK1.1

### pos

protected int **pos**

The position within the pushback buffer from which the next byte will be read. When the buffer is empty, pos is equal to buf.length; when the buffer is full, pos is equal to zero.

**Since:** JDK1.1

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

### PushbackInputStream

public **PushbackInputStream**([InputStream](http://docs.google.com/java/io/InputStream.html) in,  
 int size)

Creates a PushbackInputStream with a pushback buffer of the specified size, and saves its argument, the input stream in, for later use. Initially, there is no pushed-back byte (the field pushBack is initialized to -1).

**Parameters:**in - the input stream from which bytes will be read.size - the size of the pushback buffer. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if size is <= 0**Since:** JDK1.1

### PushbackInputStream

public **PushbackInputStream**([InputStream](http://docs.google.com/java/io/InputStream.html) in)

Creates a PushbackInputStream and saves its argument, the input stream in, for later use. Initially, there is no pushed-back byte (the field pushBack is initialized to -1).

**Parameters:**in - the input stream from which bytes will be read.

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

### 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 value 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 returns the most recently pushed-back byte, if there is one, and otherwise calls the read method of its underlying input stream and returns whatever value that method returns.

**Overrides:**[read](http://docs.google.com/java/io/FilterInputStream.html#read()) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.html) **Returns:**the next byte of data, or -1 if the end of the stream has been reached. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method, or an I/O error occurs.**See Also:**[InputStream.read()](http://docs.google.com/java/io/InputStream.html#read())

### 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. This method first reads any pushed-back bytes; after that, if fewer than len bytes have been read then it reads from the underlying input stream. 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.

**Overrides:**[read](http://docs.google.com/java/io/FilterInputStream.html#read(byte%5B%5D,%20int,%20int)) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.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 this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method, or 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))

### unread

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

Pushes back a byte by copying it to the front of the pushback buffer. After this method returns, the next byte to be read will have the value (byte)b.

**Parameters:**b - the int value whose low-order byte is to be pushed back. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If there is not enough room in the pushback buffer for the byte, or this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method.

### unread

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

Pushes back a portion of an array of bytes by copying it to the front of the pushback buffer. After this method returns, the next byte to be read will have the value b[off], the byte after that will have the value b[off+1], and so forth.

**Parameters:**b - the byte array to push back.off - the start offset of the data.len - the number of bytes to push back. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If there is not enough room in the pushback buffer for the specified number of bytes, or this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method.**Since:** JDK1.1

### unread

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

Pushes back an array of bytes by copying it to the front of the pushback buffer. After this method returns, the next byte to be read will have the value b[0], the byte after that will have the value b[1], and so forth.

**Parameters:**b - the byte array to push back **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If there is not enough room in the pushback buffer for the specified number of bytes, or this input stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method.**Since:** JDK1.1

### 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 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.

The method returns the sum of the number of bytes that have been pushed back and the value returned by [available](http://docs.google.com/java/io/FilterInputStream.html#available()).

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

### 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 this input stream. The skip method may, for a variety of reasons, end up skipping over some smaller number of bytes, possibly zero. If n is negative, no bytes are skipped.

The skip method of PushbackInputStream first skips over the bytes in the pushback buffer, if any. It then calls the skip method of the underlying input stream if more bytes need to be skipped. The actual number of bytes skipped is returned.

**Overrides:**[skip](http://docs.google.com/java/io/FilterInputStream.html#skip(long)) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.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 the stream does not support seek, or the stream has been closed by invoking its [close()](http://docs.google.com/java/io/PushbackInputStream.html#close()) method, or an I/O error occurs.**Since:** 1.2 **See Also:**[FilterInputStream.in](http://docs.google.com/java/io/FilterInputStream.html#in), [InputStream.skip(long n)](http://docs.google.com/java/io/InputStream.html#skip(long))

### markSupported

public boolean **markSupported**()

Tests if this input stream supports the mark and reset methods, which it does not.

**Overrides:**[markSupported](http://docs.google.com/java/io/FilterInputStream.html#markSupported()) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.html) **Returns:**false, since this class does not support the mark and reset methods.**See Also:**[InputStream.mark(int)](http://docs.google.com/java/io/InputStream.html#mark(int)), [InputStream.reset()](http://docs.google.com/java/io/InputStream.html#reset())

### mark

public void **mark**(int readlimit)

Marks the current position in this input stream.

The mark method of PushbackInputStream does nothing.

**Overrides:**[mark](http://docs.google.com/java/io/FilterInputStream.html#mark(int)) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.html) **Parameters:**readlimit - the maximum limit of bytes that can be read before the mark position becomes invalid.**See Also:**[InputStream.reset()](http://docs.google.com/java/io/InputStream.html#reset())

### reset

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

Repositions this stream to the position at the time the mark method was last called on this input stream.

The method reset for class PushbackInputStream does nothing except throw an IOException.

**Overrides:**[reset](http://docs.google.com/java/io/FilterInputStream.html#reset()) in class [FilterInputStream](http://docs.google.com/java/io/FilterInputStream.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if this method is invoked.**See Also:**[InputStream.mark(int)](http://docs.google.com/java/io/InputStream.html#mark(int)), [IOException](http://docs.google.com/java/io/IOException.html)

### 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. Once the stream has been closed, further read(), unread(), available(), reset(), or skip() invocations will throw an IOException. Closing a previously closed stream has no effect.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PushbackInputStream.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/PrintWriter.html)   [**NEXT CLASS**](http://docs.google.com/java/io/PushbackReader.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/PushbackInputStream.html)    [**NO FRAMES**](http://docs.google.com/PushbackInputStream.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

[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).