| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BufferedReader.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/BufferedOutputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/BufferedWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/BufferedReader.html)    [**NO FRAMES**](http://docs.google.com/BufferedReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

## **java.io**

Class BufferedReader

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

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Readable](http://docs.google.com/java/lang/Readable.html) **Direct Known Subclasses:** [LineNumberReader](http://docs.google.com/java/io/LineNumberReader.html)

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

Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

The buffer size may be specified, or the default size may be used. The default is large enough for most purposes.

In general, each read request made of a Reader causes a corresponding read request to be made of the underlying character or byte stream. It is therefore advisable to wrap a BufferedReader around any Reader whose read() operations may be costly, such as FileReaders and InputStreamReaders. For example,

BufferedReader in  
 = new BufferedReader(new FileReader("foo.in"));

will buffer the input from the specified file. Without buffering, each invocation of read() or readLine() could cause bytes to be read from the file, converted into characters, and then returned, which can be very inefficient.

Programs that use DataInputStreams for textual input can be localized by replacing each DataInputStream with an appropriate BufferedReader.

**Since:** JDK1.1 **See Also:**[FileReader](http://docs.google.com/java/io/FileReader.html), [InputStreamReader](http://docs.google.com/java/io/InputStreamReader.html)

| **Field Summary** | |
| --- | --- |

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

| **Constructor Summary** | |
| --- | --- |
| [**BufferedReader**](http://docs.google.com/java/io/BufferedReader.html#BufferedReader(java.io.Reader))([Reader](http://docs.google.com/java/io/Reader.html) in)            Creates a buffering character-input stream that uses a default-sized input buffer. |
| [**BufferedReader**](http://docs.google.com/java/io/BufferedReader.html#BufferedReader(java.io.Reader,%20int))([Reader](http://docs.google.com/java/io/Reader.html) in, int sz)            Creates a buffering character-input stream that uses an input buffer of the specified size. |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/io/BufferedReader.html#close())()            Closes the stream and releases any system resources associated with it. |
| void | [**mark**](http://docs.google.com/java/io/BufferedReader.html#mark(int))(int readAheadLimit)            Marks the present position in the stream. |
| boolean | [**markSupported**](http://docs.google.com/java/io/BufferedReader.html#markSupported())()            Tells whether this stream supports the mark() operation, which it does. |
| int | [**read**](http://docs.google.com/java/io/BufferedReader.html#read())()            Reads a single character. |
| int | [**read**](http://docs.google.com/java/io/BufferedReader.html#read(char%5B%5D,%20int,%20int))(char[] cbuf, int off, int len)            Reads characters into a portion of an array. |
| [String](http://docs.google.com/java/lang/String.html) | [**readLine**](http://docs.google.com/java/io/BufferedReader.html#readLine())()            Reads a line of text. |
| boolean | [**ready**](http://docs.google.com/java/io/BufferedReader.html#ready())()            Tells whether this stream is ready to be read. |
| void | [**reset**](http://docs.google.com/java/io/BufferedReader.html#reset())()            Resets the stream to the most recent mark. |
| long | [**skip**](http://docs.google.com/java/io/BufferedReader.html#skip(long))(long n)            Skips characters. |

| **Methods inherited from class java.io.**[**Reader**](http://docs.google.com/java/io/Reader.html) |
| --- |
| [read](http://docs.google.com/java/io/Reader.html#read(char%5B%5D)), [read](http://docs.google.com/java/io/Reader.html#read(java.nio.CharBuffer)) |

| **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** |
| --- |

### BufferedReader

public **BufferedReader**([Reader](http://docs.google.com/java/io/Reader.html) in,  
 int sz)

Creates a buffering character-input stream that uses an input buffer of the specified size.

**Parameters:**in - A Readersz - Input-buffer size **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If sz is <= 0

### BufferedReader

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

Creates a buffering character-input stream that uses a default-sized input buffer.

**Parameters:**in - A Reader

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

### read

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

Reads a single character.

**Overrides:**[read](http://docs.google.com/java/io/Reader.html#read()) in class [Reader](http://docs.google.com/java/io/Reader.html) **Returns:**The character read, as an integer in the range 0 to 65535 (0x00-0xffff), or -1 if the end of the stream has been reached **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### read

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

Reads characters into a portion of an array.

This method implements the general contract of the corresponding [read](http://docs.google.com/java/io/Reader.html#read(char%5B%5D,%20int,%20int)) method of the [Reader](http://docs.google.com/java/io/Reader.html) class. As an additional convenience, it attempts to read as many characters as possible by repeatedly invoking the read method of the underlying stream. This iterated read continues until one of the following conditions becomes true:

* The specified number of characters have been read,
* The read method of the underlying stream returns -1, indicating end-of-file, or
* The ready method of the underlying stream returns false, indicating that further input requests would block.

If the first read on the underlying stream returns -1 to indicate end-of-file then this method returns -1. Otherwise this method returns the number of characters actually read.

Subclasses of this class are encouraged, but not required, to attempt to read as many characters as possible in the same fashion.

Ordinarily this method takes characters from this stream's character buffer, filling it from the underlying stream as necessary. If, however, the buffer is empty, the mark is not valid, and the requested length is at least as large as the buffer, then this method will read characters directly from the underlying stream into the given array. Thus redundant BufferedReaders will not copy data unnecessarily.

**Specified by:**[read](http://docs.google.com/java/io/Reader.html#read(char%5B%5D,%20int,%20int)) in class [Reader](http://docs.google.com/java/io/Reader.html) **Parameters:**cbuf - Destination bufferoff - Offset at which to start storing characterslen - Maximum number of characters to read **Returns:**The number of characters read, or -1 if the end of the stream has been reached **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### readLine

public [String](http://docs.google.com/java/lang/String.html) **readLine**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads a line of text. A line is considered to be terminated by any one of a line feed ('\n'), a carriage return ('\r'), or a carriage return followed immediately by a linefeed.

**Returns:**A String containing the contents of the line, not including any line-termination characters, or null if the end of the stream has been reached **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### skip

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

Skips characters.

**Overrides:**[skip](http://docs.google.com/java/io/Reader.html#skip(long)) in class [Reader](http://docs.google.com/java/io/Reader.html) **Parameters:**n - The number of characters to skip **Returns:**The number of characters actually skipped **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If n is negative. [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### ready

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

Tells whether this stream is ready to be read. A buffered character stream is ready if the buffer is not empty, or if the underlying character stream is ready.

**Overrides:**[ready](http://docs.google.com/java/io/Reader.html#ready()) in class [Reader](http://docs.google.com/java/io/Reader.html) **Returns:**True if the next read() is guaranteed not to block for input, false otherwise. Note that returning false does not guarantee that the next read will block. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### markSupported

public boolean **markSupported**()

Tells whether this stream supports the mark() operation, which it does.

**Overrides:**[markSupported](http://docs.google.com/java/io/Reader.html#markSupported()) in class [Reader](http://docs.google.com/java/io/Reader.html) **Returns:**true if and only if this stream supports the mark operation.

### mark

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

Marks the present position in the stream. Subsequent calls to reset() will attempt to reposition the stream to this point.

**Overrides:**[mark](http://docs.google.com/java/io/Reader.html#mark(int)) in class [Reader](http://docs.google.com/java/io/Reader.html) **Parameters:**readAheadLimit - Limit on the number of characters that may be read while still preserving the mark. An attempt to reset the stream after reading characters up to this limit or beyond may fail. A limit value larger than the size of the input buffer will cause a new buffer to be allocated whose size is no smaller than limit. Therefore large values should be used with care. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If readAheadLimit is < 0 [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### reset

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

Resets the stream to the most recent mark.

**Overrides:**[reset](http://docs.google.com/java/io/Reader.html#reset()) in class [Reader](http://docs.google.com/java/io/Reader.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If the stream has never been marked, or if the mark has been invalidated

### close

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

**Description copied from class:** [**Reader**](http://docs.google.com/java/io/Reader.html#close()) Closes the stream and releases any system resources associated with it. Once the stream has been closed, further read(), ready(), mark(), 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)**Specified by:**[close](http://docs.google.com/java/io/Reader.html#close()) in class [Reader](http://docs.google.com/java/io/Reader.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/BufferedReader.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/BufferedOutputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/BufferedWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/BufferedReader.html)    [**NO FRAMES**](http://docs.google.com/BufferedReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

[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).