| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BufferedWriter.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/BufferedReader.html)   [**NEXT CLASS**](http://docs.google.com/java/io/ByteArrayInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/BufferedWriter.html)    [**NO FRAMES**](http://docs.google.com/BufferedWriter.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 BufferedWriter

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

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

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

Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.

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

A newLine() method is provided, which uses the platform's own notion of line separator as defined by the system property line.separator. Not all platforms use the newline character ('\n') to terminate lines. Calling this method to terminate each output line is therefore preferred to writing a newline character directly.

In general, a Writer sends its output immediately to the underlying character or byte stream. Unless prompt output is required, it is advisable to wrap a BufferedWriter around any Writer whose write() operations may be costly, such as FileWriters and OutputStreamWriters. For example,

PrintWriter out  
 = new PrintWriter(new BufferedWriter(new FileWriter("foo.out")));

will buffer the PrintWriter's output to the file. Without buffering, each invocation of a print() method would cause characters to be converted into bytes that would then be written immediately to the file, which can be very inefficient.

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

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

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

| **Constructor Summary** | |
| --- | --- |
| [**BufferedWriter**](http://docs.google.com/java/io/BufferedWriter.html#BufferedWriter(java.io.Writer))([Writer](http://docs.google.com/java/io/Writer.html) out)            Creates a buffered character-output stream that uses a default-sized output buffer. |
| [**BufferedWriter**](http://docs.google.com/java/io/BufferedWriter.html#BufferedWriter(java.io.Writer,%20int))([Writer](http://docs.google.com/java/io/Writer.html) out, int sz)            Creates a new buffered character-output stream that uses an output buffer of the given size. |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/io/BufferedWriter.html#close())()            Closes the stream, flushing it first. |
| void | [**flush**](http://docs.google.com/java/io/BufferedWriter.html#flush())()            Flushes the stream. |
| void | [**newLine**](http://docs.google.com/java/io/BufferedWriter.html#newLine())()            Writes a line separator. |
| void | [**write**](http://docs.google.com/java/io/BufferedWriter.html#write(char%5B%5D,%20int,%20int))(char[] cbuf, int off, int len)            Writes a portion of an array of characters. |
| void | [**write**](http://docs.google.com/java/io/BufferedWriter.html#write(int))(int c)            Writes a single character. |
| void | [**write**](http://docs.google.com/java/io/BufferedWriter.html#write(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) s, int off, int len)            Writes a portion of a String. |

| **Methods inherited from class java.io.**[**Writer**](http://docs.google.com/java/io/Writer.html) |
| --- |
| [append](http://docs.google.com/java/io/Writer.html#append(char)), [append](http://docs.google.com/java/io/Writer.html#append(java.lang.CharSequence)), [append](http://docs.google.com/java/io/Writer.html#append(java.lang.CharSequence,%20int,%20int)), [write](http://docs.google.com/java/io/Writer.html#write(char%5B%5D)), [write](http://docs.google.com/java/io/Writer.html#write(java.lang.String)) |

| **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** |
| --- |

### BufferedWriter

public **BufferedWriter**([Writer](http://docs.google.com/java/io/Writer.html) out)

Creates a buffered character-output stream that uses a default-sized output buffer.

**Parameters:**out - A Writer

### BufferedWriter

public **BufferedWriter**([Writer](http://docs.google.com/java/io/Writer.html) out,  
 int sz)

Creates a new buffered character-output stream that uses an output buffer of the given size.

**Parameters:**out - A Writersz - Output-buffer size, a positive integer **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If sz is <= 0

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

### write

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

Writes a single character.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**c - int specifying a character to be written **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### write

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

Writes a portion of an array of characters.

Ordinarily this method stores characters from the given array into this stream's buffer, flushing the buffer to the underlying stream as needed. If the requested length is at least as large as the buffer, however, then this method will flush the buffer and write the characters directly to the underlying stream. Thus redundant BufferedWriters will not copy data unnecessarily.

**Specified by:**[write](http://docs.google.com/java/io/Writer.html#write(char%5B%5D,%20int,%20int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**cbuf - A character arrayoff - Offset from which to start reading characterslen - Number of characters to write **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### write

public void **write**([String](http://docs.google.com/java/lang/String.html) s,  
 int off,  
 int len)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes a portion of a String.

If the value of the len parameter is negative then no characters are written. This is contrary to the specification of this method in the [superclass](http://docs.google.com/java/io/Writer.html#write(java.lang.String,%20int,%20int)), which requires that an [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) be thrown.

**Overrides:**[write](http://docs.google.com/java/io/Writer.html#write(java.lang.String,%20int,%20int)) in class [Writer](http://docs.google.com/java/io/Writer.html) **Parameters:**s - String to be writtenoff - Offset from which to start reading characterslen - Number of characters to be written **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### newLine

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

Writes a line separator. The line separator string is defined by the system property line.separator, and is not necessarily a single newline ('\n') character.

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

### flush

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

Flushes the stream.

**Specified by:**[flush](http://docs.google.com/java/io/Flushable.html#flush()) in interface [Flushable](http://docs.google.com/java/io/Flushable.html)**Specified by:**[flush](http://docs.google.com/java/io/Writer.html#flush()) in class [Writer](http://docs.google.com/java/io/Writer.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### close

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

**Description copied from class:** [**Writer**](http://docs.google.com/java/io/Writer.html#close()) Closes the stream, flushing it first. Once the stream has been closed, further write() or flush() invocations will cause an IOException to be thrown. 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/Writer.html#close()) in class [Writer](http://docs.google.com/java/io/Writer.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/BufferedWriter.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/BufferedReader.html)   [**NEXT CLASS**](http://docs.google.com/java/io/ByteArrayInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/BufferedWriter.html)    [**NO FRAMES**](http://docs.google.com/BufferedWriter.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).