| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Reader.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/RandomAccessFile.html)   [**NEXT CLASS**](http://docs.google.com/java/io/SequenceInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/Reader.html)    [**NO FRAMES**](http://docs.google.com/Reader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

## **java.io**

Class Reader

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

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Readable](http://docs.google.com/java/lang/Readable.html) **Direct Known Subclasses:** [BufferedReader](http://docs.google.com/java/io/BufferedReader.html), [CharArrayReader](http://docs.google.com/java/io/CharArrayReader.html), [FilterReader](http://docs.google.com/java/io/FilterReader.html), [InputStreamReader](http://docs.google.com/java/io/InputStreamReader.html), [PipedReader](http://docs.google.com/java/io/PipedReader.html), [StringReader](http://docs.google.com/java/io/StringReader.html)

public abstract class **Reader**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Readable](http://docs.google.com/java/lang/Readable.html), [Closeable](http://docs.google.com/java/io/Closeable.html)

Abstract class for reading character streams. The only methods that a subclass must implement are read(char[], int, int) and close(). Most subclasses, however, will override some of the methods defined here in order to provide higher efficiency, additional functionality, or both.

**Since:** JDK1.1 **See Also:**[BufferedReader](http://docs.google.com/java/io/BufferedReader.html), [LineNumberReader](http://docs.google.com/java/io/LineNumberReader.html), [CharArrayReader](http://docs.google.com/java/io/CharArrayReader.html), [InputStreamReader](http://docs.google.com/java/io/InputStreamReader.html), [FileReader](http://docs.google.com/java/io/FileReader.html), [FilterReader](http://docs.google.com/java/io/FilterReader.html), [PushbackReader](http://docs.google.com/java/io/PushbackReader.html), [PipedReader](http://docs.google.com/java/io/PipedReader.html), [StringReader](http://docs.google.com/java/io/StringReader.html), [Writer](http://docs.google.com/java/io/Writer.html)

| **Field Summary** | |
| --- | --- |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**lock**](http://docs.google.com/java/io/Reader.html#lock)            The object used to synchronize operations on this stream. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Reader**](http://docs.google.com/java/io/Reader.html#Reader())()            Creates a new character-stream reader whose critical sections will synchronize on the reader itself. |
| protected | [**Reader**](http://docs.google.com/java/io/Reader.html#Reader(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) lock)            Creates a new character-stream reader whose critical sections will synchronize on the given object. |

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

| **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** |
| --- |

### lock

protected [Object](http://docs.google.com/java/lang/Object.html) **lock**

The object used to synchronize operations on this stream. For efficiency, a character-stream object may use an object other than itself to protect critical sections. A subclass should therefore use the object in this field rather than this or a synchronized method.

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

### Reader

protected **Reader**()

Creates a new character-stream reader whose critical sections will synchronize on the reader itself.

### Reader

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

Creates a new character-stream reader whose critical sections will synchronize on the given object.

**Parameters:**lock - The Object to synchronize on.

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

### read

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

Attempts to read characters into the specified character buffer. The buffer is used as a repository of characters as-is: the only changes made are the results of a put operation. No flipping or rewinding of the buffer is performed.

**Specified by:**[read](http://docs.google.com/java/lang/Readable.html#read(java.nio.CharBuffer)) in interface [Readable](http://docs.google.com/java/lang/Readable.html) **Parameters:**target - the buffer to read characters into **Returns:**The number of characters added to the buffer, or -1 if this source of characters is at its end **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if target is null ReadOnlyBufferException - if target is a read only buffer**Since:** 1.5

### read

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

Reads a single character. This method will block until a character is available, an I/O error occurs, or the end of the stream is reached.

Subclasses that intend to support efficient single-character input should override this method.

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

Reads characters into an array. This method will block until some input is available, an I/O error occurs, or the end of the stream is reached.

**Parameters:**cbuf - Destination buffer **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

### read

public abstract 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 will block until some input is available, an I/O error occurs, or the end of the stream is reached.

**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

### skip

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

Skips characters. This method will block until some characters are available, an I/O error occurs, or the end of the stream is reached.

**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.

**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. The default implementation always returns false. Subclasses should override this method.

**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. Not all character-input streams support the mark() operation.

**Parameters:**readAheadLimit - Limit on the number of characters that may be read while still preserving the mark. After reading this many characters, attempting to reset the stream may fail. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If the stream does not support mark(), or if some other I/O error occurs

### reset

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

Resets the stream. If the stream has been marked, then attempt to reposition it at the mark. If the stream has not been marked, then attempt to reset it in some way appropriate to the particular stream, for example by repositioning it to its starting point. Not all character-input streams support the reset() operation, and some support reset() without supporting mark().

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If the stream has not been marked, or if the mark has been invalidated, or if the stream does not support reset(), or if some other I/O error occurs

### close

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

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) **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/Reader.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/RandomAccessFile.html)   [**NEXT CLASS**](http://docs.google.com/java/io/SequenceInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/Reader.html)    [**NO FRAMES**](http://docs.google.com/Reader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [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).