| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CharArrayReader.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/ByteArrayOutputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/CharArrayWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/CharArrayReader.html)    [**NO FRAMES**](http://docs.google.com/CharArrayReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#2jxsxqh) |

## **java.io**

Class CharArrayReader

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

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

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

This class implements a character buffer that can be used as a character-input stream.

**Since:** JDK1.1

| **Field Summary** | |
| --- | --- |
| protected  char[] | [**buf**](http://docs.google.com/java/io/CharArrayReader.html#buf)            The character buffer. |
| protected  int | [**count**](http://docs.google.com/java/io/CharArrayReader.html#count)            The index of the end of this buffer. |
| protected  int | [**markedPos**](http://docs.google.com/java/io/CharArrayReader.html#markedPos)            The position of mark in buffer. |
| protected  int | [**pos**](http://docs.google.com/java/io/CharArrayReader.html#pos)            The current buffer position. |

| **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** | |
| --- | --- |
| [**CharArrayReader**](http://docs.google.com/java/io/CharArrayReader.html#CharArrayReader(char%5B%5D))(char[] buf)            Creates a CharArrayReader from the specified array of chars. |
| [**CharArrayReader**](http://docs.google.com/java/io/CharArrayReader.html#CharArrayReader(char%5B%5D,%20int,%20int))(char[] buf, int offset, int length)            Creates a CharArrayReader from the specified array of chars. |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/io/CharArrayReader.html#close())()            Closes the stream and releases any system resources associated with it. |
| void | [**mark**](http://docs.google.com/java/io/CharArrayReader.html#mark(int))(int readAheadLimit)            Marks the present position in the stream. |
| boolean | [**markSupported**](http://docs.google.com/java/io/CharArrayReader.html#markSupported())()            Tells whether this stream supports the mark() operation, which it does. |
| int | [**read**](http://docs.google.com/java/io/CharArrayReader.html#read())()            Reads a single character. |
| int | [**read**](http://docs.google.com/java/io/CharArrayReader.html#read(char%5B%5D,%20int,%20int))(char[] b, int off, int len)            Reads characters into a portion of an array. |
| boolean | [**ready**](http://docs.google.com/java/io/CharArrayReader.html#ready())()            Tells whether this stream is ready to be read. |
| void | [**reset**](http://docs.google.com/java/io/CharArrayReader.html#reset())()            Resets the stream to the most recent mark, or to the beginning if it has never been marked. |
| long | [**skip**](http://docs.google.com/java/io/CharArrayReader.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)) |

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

### buf

protected char[] **buf**

The character buffer.

### pos

protected int **pos**

The current buffer position.

### markedPos

protected int **markedPos**

The position of mark in buffer.

### count

protected int **count**

The index of the end of this buffer. There is not valid data at or beyond this index.

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

### CharArrayReader

public **CharArrayReader**(char[] buf)

Creates a CharArrayReader from the specified array of chars.

**Parameters:**buf - Input buffer (not copied)

### CharArrayReader

public **CharArrayReader**(char[] buf,  
 int offset,  
 int length)

Creates a CharArrayReader from the specified array of chars.

The resulting reader will start reading at the given offset. The total number of char values that can be read from this reader will be either length or buf.length-offset, whichever is smaller.

**Parameters:**buf - Input buffer (not copied)offset - Offset of the first char to readlength - Number of chars to read **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If offset is negative or greater than buf.length, or if length is negative, or if the sum of these two values is negative.

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

Reads characters into a portion of an array.

**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:**b - Destination bufferoff - Offset at which to start storing characterslen - Maximum number of characters to read **Returns:**The actual 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. Returns the number of characters that were skipped.

The n parameter may be negative, even though the skip method of the [Reader](http://docs.google.com/java/io/Reader.html) superclass throws an exception in this case. If n is negative, then this method does nothing and returns 0.

**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:** [IOException](http://docs.google.com/java/io/IOException.html) - If the stream is closed, or 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. Character-array readers are always ready to be read.

**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 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. Because the stream's input comes from a character array, there is no actual limit; hence this argument is ignored. **Throws:** [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, or to the beginning if it has never been marked.

**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 an I/O error occurs

### close

public void **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)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CharArrayReader.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/ByteArrayOutputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/CharArrayWriter.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/CharArrayReader.html)    [**NO FRAMES**](http://docs.google.com/CharArrayReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#2jxsxqh) |

[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).