| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LineNumberReader.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/LineNumberInputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/NotActiveException.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/LineNumberReader.html)    [**NO FRAMES**](http://docs.google.com/LineNumberReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#35nkun2) |

## **java.io**

Class LineNumberReader

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

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

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

A buffered character-input stream that keeps track of line numbers. This class defines methods [setLineNumber(int)](http://docs.google.com/java/io/LineNumberReader.html#setLineNumber(int)) and [getLineNumber()](http://docs.google.com/java/io/LineNumberReader.html#getLineNumber()) for setting and getting the current line number respectively.

By default, line numbering begins at 0. This number increments at every [line terminator](#3znysh7) as the data is read, and can be changed with a call to setLineNumber(int). Note however, that setLineNumber(int) does not actually change the current position in the stream; it only changes the value that will be returned by getLineNumber().

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.

**Since:** JDK1.1

| **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** | |
| --- | --- |
| [**LineNumberReader**](http://docs.google.com/java/io/LineNumberReader.html#LineNumberReader(java.io.Reader))([Reader](http://docs.google.com/java/io/Reader.html) in)            Create a new line-numbering reader, using the default input-buffer size. |
| [**LineNumberReader**](http://docs.google.com/java/io/LineNumberReader.html#LineNumberReader(java.io.Reader,%20int))([Reader](http://docs.google.com/java/io/Reader.html) in, int sz)            Create a new line-numbering reader, reading characters into a buffer of the given size. |

| **Method Summary** | |
| --- | --- |
| int | [**getLineNumber**](http://docs.google.com/java/io/LineNumberReader.html#getLineNumber())()            Get the current line number. |
| void | [**mark**](http://docs.google.com/java/io/LineNumberReader.html#mark(int))(int readAheadLimit)            Mark the present position in the stream. |
| int | [**read**](http://docs.google.com/java/io/LineNumberReader.html#read())()            Read a single character. |
| int | [**read**](http://docs.google.com/java/io/LineNumberReader.html#read(char%5B%5D,%20int,%20int))(char[] cbuf, int off, int len)            Read characters into a portion of an array. |
| [String](http://docs.google.com/java/lang/String.html) | [**readLine**](http://docs.google.com/java/io/LineNumberReader.html#readLine())()            Read a line of text. |
| void | [**reset**](http://docs.google.com/java/io/LineNumberReader.html#reset())()            Reset the stream to the most recent mark. |
| void | [**setLineNumber**](http://docs.google.com/java/io/LineNumberReader.html#setLineNumber(int))(int lineNumber)            Set the current line number. |
| long | [**skip**](http://docs.google.com/java/io/LineNumberReader.html#skip(long))(long n)            Skip characters. |

| **Methods inherited from class java.io.**[**BufferedReader**](http://docs.google.com/java/io/BufferedReader.html) |
| --- |
| [close](http://docs.google.com/java/io/BufferedReader.html#close()), [markSupported](http://docs.google.com/java/io/BufferedReader.html#markSupported()), [ready](http://docs.google.com/java/io/BufferedReader.html#ready()) |

| **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** |
| --- |

### LineNumberReader

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

Create a new line-numbering reader, using the default input-buffer size.

**Parameters:**in - A Reader object to provide the underlying stream

### LineNumberReader

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

Create a new line-numbering reader, reading characters into a buffer of the given size.

**Parameters:**in - A Reader object to provide the underlying streamsz - An int specifying the size of the buffer

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

### setLineNumber

public void **setLineNumber**(int lineNumber)

Set the current line number.

**Parameters:**lineNumber - An int specifying the line number**See Also:**[getLineNumber()](http://docs.google.com/java/io/LineNumberReader.html#getLineNumber())

### getLineNumber

public int **getLineNumber**()

Get the current line number.

**Returns:**The current line number**See Also:**[setLineNumber(int)](http://docs.google.com/java/io/LineNumberReader.html#setLineNumber(int))

### read

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

Read a single character. [Line terminators](#3znysh7) are compressed into single newline ('\n') characters. Whenever a line terminator is read the current line number is incremented.

**Overrides:**[read](http://docs.google.com/java/io/BufferedReader.html#read()) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.html) **Returns:**The character 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 int **read**(char[] cbuf,  
 int off,  
 int len)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Read characters into a portion of an array. Whenever a [line terminator](#3znysh7) is read the current line number is incremented.

**Overrides:**[read](http://docs.google.com/java/io/BufferedReader.html#read(char%5B%5D,%20int,%20int)) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.html) **Parameters:**cbuf - Destination bufferoff - Offset at which to start storing characterslen - Maximum number of characters to read **Returns:**The number of bytes read, or -1 if the end of the stream has already 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)

Read a line of text. Whenever a [line terminator](#3znysh7) is read the current line number is incremented.

**Overrides:**[readLine](http://docs.google.com/java/io/BufferedReader.html#readLine()) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.html) **Returns:**A String containing the contents of the line, not including any [line termination characters](#3znysh7), 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)

Skip characters.

**Overrides:**[skip](http://docs.google.com/java/io/BufferedReader.html#skip(long)) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.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 an I/O error occurs [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If n is negative

### mark

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

Mark the present position in the stream. Subsequent calls to reset() will attempt to reposition the stream to this point, and will also reset the line number appropriately.

**Overrides:**[mark](http://docs.google.com/java/io/BufferedReader.html#mark(int)) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.html) **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 an I/O error occurs

### reset

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

Reset the stream to the most recent mark.

**Overrides:**[reset](http://docs.google.com/java/io/BufferedReader.html#reset()) in class [BufferedReader](http://docs.google.com/java/io/BufferedReader.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If the stream has not been marked, or if the mark has been invalidated

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LineNumberReader.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/LineNumberInputStream.html)   [**NEXT CLASS**](http://docs.google.com/java/io/NotActiveException.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/LineNumberReader.html)    [**NO FRAMES**](http://docs.google.com/LineNumberReader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#35nkun2) |

[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).