| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Console.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/Closeable.html)   [**NEXT CLASS**](http://docs.google.com/java/io/DataInput.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/Console.html)    [**NO FRAMES**](http://docs.google.com/Console.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.io**

Class Console

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

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

public final class **Console**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Flushable](http://docs.google.com/java/io/Flushable.html)

Methods to access the character-based console device, if any, associated with the current Java virtual machine.

Whether a virtual machine has a console is dependent upon the underlying platform and also upon the manner in which the virtual machine is invoked. If the virtual machine is started from an interactive command line without redirecting the standard input and output streams then its console will exist and will typically be connected to the keyboard and display from which the virtual machine was launched. If the virtual machine is started automatically, for example by a background job scheduler, then it will typically not have a console.

If this virtual machine has a console then it is represented by a unique instance of this class which can be obtained by invoking the [System.console()](http://docs.google.com/java/lang/System.html#console()) method. If no console device is available then an invocation of that method will return null.

Read and write operations are synchronized to guarantee the atomic completion of critical operations; therefore invoking methods [readLine()](http://docs.google.com/java/io/Console.html#readLine()), [readPassword()](http://docs.google.com/java/io/Console.html#readPassword()), [format()](http://docs.google.com/java/io/Console.html#format(java.lang.String,%20java.lang.Object...)), [printf()](http://docs.google.com/java/io/Console.html#printf(java.lang.String,%20java.lang.Object...)) as well as the read, format and write operations on the objects returned by [reader()](http://docs.google.com/java/io/Console.html#reader()) and [writer()](http://docs.google.com/java/io/Console.html#writer()) may block in multithreaded scenarios.

Invoking close() on the objects returned by the [reader()](http://docs.google.com/java/io/Console.html#reader()) and the [writer()](http://docs.google.com/java/io/Console.html#writer()) will not close the underlying stream of those objects.

The console-read methods return null when the end of the console input stream is reached, for example by typing control-D on Unix or control-Z on Windows. Subsequent read operations will succeed if additional characters are later entered on the console's input device.

Unless otherwise specified, passing a null argument to any method in this class will cause a [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) to be thrown.

**Security note:** If an application needs to read a password or other secure data, it should use [readPassword()](http://docs.google.com/java/io/Console.html#readPassword()) or [readPassword(String, Object...)](http://docs.google.com/java/io/Console.html#readPassword(java.lang.String,%20java.lang.Object...)) and manually zero the returned character array after processing to minimize the lifetime of sensitive data in memory.

Console cons;  
 char[] passwd;  
 if ((cons = System.console()) != null &&  
 (passwd = cons.readPassword("[%s]", "Password:")) != null) {  
 ...  
 java.util.Arrays.fill(passwd, ' ');  
 }

**Since:** 1.6

| **Method Summary** | |
| --- | --- |
| void | [**flush**](http://docs.google.com/java/io/Console.html#flush())()            Flushes the console and forces any buffered output to be written immediately . |
| [Console](http://docs.google.com/java/io/Console.html) | [**format**](http://docs.google.com/java/io/Console.html#format(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) fmt, [Object](http://docs.google.com/java/lang/Object.html)... args)            Writes a formatted string to this console's output stream using the specified format string and arguments. |
| [Console](http://docs.google.com/java/io/Console.html) | [**printf**](http://docs.google.com/java/io/Console.html#printf(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) format, [Object](http://docs.google.com/java/lang/Object.html)... args)            A convenience method to write a formatted string to this console's output stream using the specified format string and arguments. |
| [Reader](http://docs.google.com/java/io/Reader.html) | [**reader**](http://docs.google.com/java/io/Console.html#reader())()            Retrieves the unique [Reader](http://docs.google.com/java/io/Reader.html) object associated with this console. |
| [String](http://docs.google.com/java/lang/String.html) | [**readLine**](http://docs.google.com/java/io/Console.html#readLine())()            Reads a single line of text from the console. |
| [String](http://docs.google.com/java/lang/String.html) | [**readLine**](http://docs.google.com/java/io/Console.html#readLine(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) fmt, [Object](http://docs.google.com/java/lang/Object.html)... args)            Provides a formatted prompt, then reads a single line of text from the console. |
| char[] | [**readPassword**](http://docs.google.com/java/io/Console.html#readPassword())()            Reads a password or passphrase from the console with echoing disabled |
| char[] | [**readPassword**](http://docs.google.com/java/io/Console.html#readPassword(java.lang.String,%20java.lang.Object...))([String](http://docs.google.com/java/lang/String.html) fmt, [Object](http://docs.google.com/java/lang/Object.html)... args)            Provides a formatted prompt, then reads a password or passphrase from the console with echoing disabled. |
| [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) | [**writer**](http://docs.google.com/java/io/Console.html#writer())()            Retrieves the unique [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) object associated with this console. |

| **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)) |

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

### writer

public [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) **writer**()

Retrieves the unique [PrintWriter](http://docs.google.com/java/io/PrintWriter.html) object associated with this console.

**Returns:**The printwriter associated with this console

### reader

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

Retrieves the unique [Reader](http://docs.google.com/java/io/Reader.html) object associated with this console.

This method is intended to be used by sophisticated applications, for example, a [Scanner](http://docs.google.com/java/util/Scanner.html) object which utilizes the rich parsing/scanning functionality provided by the Scanner:

Console con = System.console();  
 if (con != null) {  
 Scanner sc = new Scanner(con.reader());  
 ...  
 }

For simple applications requiring only line-oriented reading, use [readLine(java.lang.String, java.lang.Object...)](http://docs.google.com/java/io/Console.html#readLine(java.lang.String,%20java.lang.Object...)).

The bulk read operations [read(char[])](http://docs.google.com/java/io/Reader.html#read(char%5B%5D)) , [read(char[], int, int)](http://docs.google.com/java/io/Reader.html#read(char%5B%5D,%20int,%20int))  and [read(java.nio.CharBuffer)](http://docs.google.com/java/io/Reader.html#read(java.nio.CharBuffer)) on the returned object will not read in characters beyond the line bound for each invocation, even if the destination buffer has space for more characters. A line bound is considered to be any one of a line feed ('\n'), a carriage return ('\r'), a carriage return followed immediately by a linefeed, or an end of stream.

**Returns:**The reader associated with this console

### format

public [Console](http://docs.google.com/java/io/Console.html) **format**([String](http://docs.google.com/java/lang/String.html) fmt,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

Writes a formatted string to this console's output stream using the specified format string and arguments.

**Parameters:**fmt - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax)args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This console **Throws:** [IllegalFormatException](http://docs.google.com/java/util/IllegalFormatException.html) - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification.

### printf

public [Console](http://docs.google.com/java/io/Console.html) **printf**([String](http://docs.google.com/java/lang/String.html) format,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

A convenience method to write a formatted string to this console's output stream using the specified format string and arguments.

An invocation of this method of the form con.printf(format, args) behaves in exactly the same way as the invocation of

con.format(format, args)

.

**Parameters:**format - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax).args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The number of arguments is variable and may be zero. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). The behaviour on a null argument depends on the [conversion](http://docs.google.com/util/Formatter.html#syntax). **Returns:**This console **Throws:** [IllegalFormatException](http://docs.google.com/java/util/IllegalFormatException.html) - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification.

### readLine

public [String](http://docs.google.com/java/lang/String.html) **readLine**([String](http://docs.google.com/java/lang/String.html) fmt,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

Provides a formatted prompt, then reads a single line of text from the console.

**Parameters:**fmt - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax).args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). **Returns:**A string containing the line read from the console, not including any line-termination characters, or null if an end of stream has been reached. **Throws:** [IllegalFormatException](http://docs.google.com/java/util/IllegalFormatException.html) - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [IOError](http://docs.google.com/java/io/IOError.html) - If an I/O error occurs.

### readLine

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

Reads a single line of text from the console.

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

### readPassword

public char[] **readPassword**([String](http://docs.google.com/java/lang/String.html) fmt,  
 [Object](http://docs.google.com/java/lang/Object.html)... args)

Provides a formatted prompt, then reads a password or passphrase from the console with echoing disabled.

**Parameters:**fmt - A format string as described in [Format string syntax](http://docs.google.com/util/Formatter.html#syntax) for the prompt text.args - Arguments referenced by the format specifiers in the format string. If there are more arguments than format specifiers, the extra arguments are ignored. The maximum number of arguments is limited by the maximum dimension of a Java array as defined by the [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/). **Returns:**A character array containing the password or passphrase read from the console, not including any line-termination characters, or null if an end of stream has been reached. **Throws:** [IllegalFormatException](http://docs.google.com/java/util/IllegalFormatException.html) - If a format string contains an illegal syntax, a format specifier that is incompatible with the given arguments, insufficient arguments given the format string, or other illegal conditions. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification. [IOError](http://docs.google.com/java/io/IOError.html) - If an I/O error occurs.

### readPassword

public char[] **readPassword**()

Reads a password or passphrase from the console with echoing disabled

**Returns:**A character array containing the password or passphrase read from the console, not including any line-termination characters, or null if an end of stream has been reached. **Throws:** [IOError](http://docs.google.com/java/io/IOError.html) - If an I/O error occurs.

### flush

public void **flush**()

Flushes the console and forces any buffered output to be written immediately .

**Specified by:**[flush](http://docs.google.com/java/io/Flushable.html#flush()) in interface [Flushable](http://docs.google.com/java/io/Flushable.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/Console.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/Closeable.html)   [**NEXT CLASS**](http://docs.google.com/java/io/DataInput.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/Console.html)    [**NO FRAMES**](http://docs.google.com/Console.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).