

java.io Class Writer

[java.lang.Object](#)

|
+-- **java.io.Writer**

Direct Known Subclasses:

[BufferedWriter](#), [CharArrayWriter](#), [FilterWriter](#), [OutputStreamWriter](#), [PipedWriter](#),
[PrintWriter](#), [StringWriter](#)

public abstract class **Writer**
extends [Object](#)

Abstract class for writing to character streams. The only methods that a subclass must implement are `write(char[], int, int)`, `flush()`, 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:
[Writer](#), [BufferedWriter](#), [CharArrayWriter](#), [FilterWriter](#),
[OutputStreamWriter](#), [FileWriter](#), [PipedWriter](#), [PrintWriter](#),
[StringWriter](#), [Reader](#)

Field Summary

protected Object	lock The object used to synchronize operations on this stream.
-------------------------------------	---

Constructor Summary

protected	Writer () Create a new character-stream writer whose critical sections will synchronize on the writer itself.
protected	Writer (Object lock) Create a new character-stream writer whose critical sections will synchronize on the given object.

Method Summary

abstract void	close () Close the stream, flushing it first.
abstract void	flush () Flush the stream.
void	write (char[] cbuf) Write an array of characters.
abstract void	write (char[] cbuf, int off, int len) Write a portion of an array of characters.
void	write (int c) Write a single character.
void	write (String str) Write a string.
void	write (String str, int off, int len) Write a portion of a string.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

lock

protected [Object](#) **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

Writer

protected **Writer**()

Create a new character-stream writer whose critical sections will synchronize on the writer itself.

Writer

protected **Writer**([Object](#) lock)

Create a new character-stream writer whose critical sections will synchronize on the given object.

Parameters:

lock - Object to synchronize on.

Method Detail

write

```
public void write(int c)
    throws IOException
```

Write a single character. The character to be written is contained in the 16 low-order bits of the given integer value; the 16 high-order bits are ignored.

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

Parameters:

`c` - int specifying a character to be written.

Throws:

[IOException](#) - If an I/O error occurs

write

```
public void write(char[] cbuf)
    throws IOException
```

Write an array of characters.

Parameters:

`cbuf` - Array of characters to be written

Throws:

[IOException](#) - If an I/O error occurs

write

```
public abstract void write(char[] cbuf,
                           int off,
                           int len)
    throws IOException
```

Write a portion of an array of characters.

Parameters:

`cbuf` - Array of characters

`off` - Offset from which to start writing characters

`len` - Number of characters to write

Throws:

[IOException](#) - If an I/O error occurs

write

```
public void write(String str)
    throws IOException
```

Write a string.

Parameters:

`str` - String to be written

Throws:

[IOException](#) - If an I/O error occurs

write

```
public void write(String str,
                 int off,
                 int len)
    throws IOException
```

Write a portion of a string.

Parameters:

`str` - A String

`off` - Offset from which to start writing characters

`len` - Number of characters to write

Throws:

[IOException](#) - If an I/O error occurs

flush

```
public abstract void flush()
    throws IOException
```

Flush the stream. If the stream has saved any characters from the various write() methods in a buffer, write them immediately to their intended destination. Then, if that destination is another character or byte stream, flush it. Thus one flush() invocation will flush all the buffers in a chain of Writers and OutputStreams.

Throws:

[IOException](#) - If an I/O error occurs

close

```
public abstract void close()
    throws IOException
```

Close the stream, flushing it first. Once a stream has been closed, further write() or flush() invocations will cause an IOException to be thrown. Closing a previously-closed stream, however, has no effect.

Throws:

[IOException](#) - If an I/O error occurs

[Overview](#) [Package](#) **[Class](#)** [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

*Java™ 2 Platform
Std. Ed. v1.3.1*

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Submit a bug or feature](#)

For further API reference and developer documentation, see [Java 2 SDK SE Developer Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Java, Java 2D, and JDBC are trademarks or registered trademarks of Sun Microsystems, Inc. in the US and other countries.

Copyright 1993-2001 Sun Microsystems, Inc. 901 San Antonio Road
Palo Alto, California, 94303, U.S.A. All Rights Reserved.