

---

# NSOutputStream Class Reference

[Cocoa](#) > [Data Management](#)



2009-01-06



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **NSOutputStream Class Reference 5**

---

Overview	5
Subclassing Notes	5
Tasks	6
Creating Streams	6
Using Streams	6
Class Methods	6
outputStreamToBuffer:capacity:	6
outputStreamToFileAtPath:append:	7
outputStreamToMemory	8
Instance Methods	8
hasSpaceAvailable	8
initWithBuffer:capacity:	8
initWithFileAtPath:append:	9
initWithMemory	10
write:maxLength:	10

---

## **Document Revision History 11**

---

## **Index 13**

---



# NSOutputStream Class Reference

---

<b>Inherits from</b>	NSStream : NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.3 and later.
<b>Companion guide</b>	Stream Programming Guide for Cocoa
<b>Declared in</b>	NSStream.h
<b>Related sample code</b>	CocoaEcho CocoaHTTPServer CocoaSOAP

## Overview

The `NSOutputStream` class is a subclass of `NSStream` that provides write-only stream functionality.

## Subclassing Notes

---

The `NSOutputStream` is a concrete subclass of `NSStream` that lets you write data to a stream. Although `NSOutputStream` is probably sufficient for most situations requiring this capability, you can create a subclass of `NSOutputStream` if you want more specialized behavior (for example, you want to record statistics on the data in a stream).

## Methods to Override

---

To create a subclass of `NSOutputStream` you may have to implement initializers for the type of stream data supported and suitably reimplement existing initializers. You must also provide complete implementations of the following methods:

- [write:maxLength:](#) (page 10)

From the current write pointer, take up to the number of bytes specified in the `maxLength:` parameter from the client-supplied buffer (first parameter) and put them onto the stream. The buffer must be of the size specified by the second parameter. To prepare for the next operation, offset the write pointer by the number of bytes written. Return a signed integer based on the outcome of the current operation:

- If the write operation is successful, return the actual number of bytes put onto the stream.

- ❑ If there was an error writing to the stream, return -1.
- ❑ If the stream is of a fixed length and has reached its capacity, return zero.
- [hasSpaceAvailable](#) (page 8)
 

Return YES if the stream can currently accept more data, NO if it cannot. If you want to be semantically compatible with `NSOutputStream`, return YES if a write must be attempted to determine if space is available.

## Tasks

### Creating Streams

- + [outputStreamToMemory](#) (page 8)
 

Creates and returns an initialized output stream that will write stream data to memory.
- + [outputStreamToBuffer:capacity:](#) (page 6)
 

Creates and returns an initialized output stream that can write to a provided buffer.
- + [outputStreamToFileAtPath:append:](#) (page 7)
 

Creates and returns an initialized output stream for writing to a specified file.
- [initWithMemory](#) (page 10)
 

Returns an initialized output stream that will write to memory.
- [initWithBuffer:capacity:](#) (page 8)
 

Returns an initialized output stream that can write to a provided buffer.
- [initWithFileAtPath:append:](#) (page 9)
 

Returns an initialized output stream for writing to a specified file.

### Using Streams

- [hasSpaceAvailable](#) (page 8)
 

Returns whether the receiver can be written to.
- [write:maxLength:](#) (page 10)
 

Writes the contents of a provided data buffer to the receiver.

## Class Methods

### **outputStreamToBuffer:capacity:**

Creates and returns an initialized output stream that can write to a provided buffer.

```
+ (id)outputStreamToBuffer:(uint8_t *)buffer capacity:(NSUInteger)capacity
```

**Parameters***buffer*

The buffer the output stream will write to.

*capacity*

The size of the buffer in bytes.

**Return Value**An initialized output stream that can write to *buffer*.**Discussion**

The stream must be opened before it can be used.

When the number of bytes written to *buffer* has reached *capacity*, the stream's *streamStatus* will return *NSStreamStatusAtEnd*.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- + [outputStreamToMemory](#) (page 8)
- + [outputStreamToFileAtPath:append:](#) (page 7)
- [initWithBuffer:capacity:](#) (page 8)

**Declared In**

NSStream.h

**outputStreamToFileAtPath:append:**

Creates and returns an initialized output stream for writing to a specified file.

```
+ (id)outputStreamToFileAtPath:(NSString *)path append:(BOOL)shouldAppend
```

**Parameters***path*

The path to the file the output stream will write to.

*shouldAppend*

YES if newly written data should be appended to any existing file contents, NO otherwise.

**Return Value**An initialized output stream that can write to *path*.**Discussion**

The stream must be opened before it can be used.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- + [outputStreamToMemory](#) (page 8)
- + [outputStreamToBuffer:capacity:](#) (page 6)
- [initWithFileAtPath:append:](#) (page 9)

**Declared In**  
NSStream.h

## outputStreamToMemory

Creates and returns an initialized output stream that will write stream data to memory.

+ (id)outputStreamToMemory

### Return Value

An initialized output stream that will write stream data to memory.

### Discussion

The stream must be opened before it can be used.

You retrieve the contents of the memory stream by sending the message `propertyForKey:` to the receiver with an argument of `NSStreamDataWrittenToMemoryStreamKey`.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- + [outputStreamToBuffer:capacity:](#) (page 6)
- + [outputStreamToFileAtPath:append:](#) (page 7)
- [initToMemory](#) (page 10)

**Declared In**  
NSStream.h

## Instance Methods

### hasSpaceAvailable

Returns whether the receiver can be written to.

- (BOOL)hasSpaceAvailable

### Return Value

YES if the receiver can be written to or if a write must be attempted in order to determine if space is available, NO otherwise.

### Availability

Available in Mac OS X v10.3 and later.

**Declared In**  
NSStream.h

### initToBuffer:capacity:

Returns an initialized output stream that can write to a provided buffer.



- (id)initToBuffer:(uint8\_t \*)*buffer* capacity:(NSUInteger)*capacity*

#### Parameters

*buffer*

The buffer the output stream will write to.

*capacity*

The size of the buffer in bytes.

#### Return Value

An initialized output stream that can write to *buffer*.

#### Discussion

The stream must be opened before it can be used.

When the number of bytes written to *buffer* has reached *capacity*, the stream's `streamStatus` will return `NSStreamStatusAtEnd`.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [initToMemory](#) (page 10)
- [initToFileAtPath:append:](#) (page 9)
- + [outputStreamToBuffer:capacity:](#) (page 6)

#### Declared In

NSStream.h

## initToFileAtPath:append:

Returns an initialized output stream for writing to a specified file.

- (id)initToFileAtPath:(NSString \*)*path* append:(BOOL)*shouldAppend*

#### Parameters

*path*

The path to the file the output stream will write to.

*shouldAppend*

YES if newly written data should be appended to any existing file contents, NO otherwise.

#### Return Value

An initialized output stream that can write to *path*.

#### Discussion

The stream must be opened before it can be used.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [initToMemory](#) (page 10)
- [initToBuffer:capacity:](#) (page 8)
- + [outputStreamToFileAtPath:append:](#) (page 7)

**Declared In**  
 NSStream.h

## initWithMemory

Returns an initialized output stream that will write to memory.

- (id)initWithMemory

### Return Value

An initialized output stream that will write stream data to memory.

### Discussion

The stream must be opened before it can be used.

The contents of the memory stream are retrieved by passing the constant `NSStreamDataWrittenToMemoryStreamKey` to `propertyForKey:`.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [initWithBuffer:capacity:](#) (page 8)
- [initWithFileAtPath:append:](#) (page 9)
- + [outputStreamToMemory](#) (page 8)

**Declared In**  
 NSStream.h

## write:maxLength:

Writes the contents of a provided data buffer to the receiver.

- (NSInteger)write:(const uint8\_t \*)*buffer* maxLength:(NSUInteger)*length*

### Parameters

*buffer*

The data to write.

*length*

The length of the data buffer, in bytes.

### Return Value

The number of bytes actually written, or -1 if an error occurs. More information about the error can be obtained with `streamError`. If the receiver is a fixed-length stream and has reached its capacity, 0 is returned.

### Availability

Available in Mac OS X v10.3 and later.

### Related Sample Code

CocoaEcho

**Declared In**  
 NSStream.h

# Document Revision History

---

This table describes the changes to *NSOutputStream Class Reference*.

Date	Notes
2009-01-06	Miscellaneous edits.
2007-04-05	Made editorial improvements.
2006-11-07	Corrected overriding details for the <code>hasSpaceAvailable</code> method in the "Subclassing Notes" section.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## H

---

hasSpaceAvailable **instance method** [8](#)

## I

---

initWithBuffer:capacity: **instance method** [8](#)  
initWithFileAtPath:append: **instance method** [9](#)  
initWithMemory **instance method** [10](#)

## O

---

outputStreamToBuffer:capacity: **class method** [6](#)  
outputStreamToFileAtPath:append: **class method**  
[7](#)  
outputStreamToMemory **class method** [8](#)

## W

---

write:maxLength: **instance method** [10](#)