

---

# NSInputStream Class Reference

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



2007-04-05



Apple Inc.  
© 2007 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

## **InputStream Class Reference 5**

---

- Overview 5
  - Subclassing Notes 5
- Tasks 6
  - Creating Streams 6
  - Using Streams 6
- Class Methods 6
  - inputStreamWithData: 6
  - inputStreamWithFileAtPath: 7
- Instance Methods 7
  - getBuffer:length: 7
  - hasBytesAvailable 8
  - initWithData: 8
  - initWithFileAtPath: 9
  - read:maxLength: 9

## **Document Revision History 11**

---

## **Index 13**

---



# InputStream 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

`InputStream` is a subclass of `NSStream` that provides read-only stream functionality.

## Subclassing Notes

---

`InputStream` is a concrete subclass of `NSStream` that gives you standard read-only access to stream data. Although `InputStream` is probably sufficient for most situations requiring access to stream data, you can create a subclass of `InputStream` 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 `InputStream` 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:

- `read:maxLength:` (page 9)

From the current read index, take up to the number of bytes specified in the second parameter from the stream and place them in the client-supplied buffer (first parameter). The buffer must be of the size specified by the second parameter. Return the actual number of bytes placed in the buffer; if there is nothing left in the stream, return 0. Reset the index into the stream for the next read operation.

- `getBuffer:length:` (page 7)

Return in 0(1) a pointer to the subclass-allocated buffer (first parameter). Return by reference in the second parameter the number of bytes actually put into the buffer. The buffer's contents are valid only until the next stream operation. Return NO if you cannot access data in the buffer; otherwise, return YES. If this method is not appropriate for your type of stream, you may return NO.

■ [hasBytesAvailable](#) (page 8)

Return YES if there is more data to read in the stream, NO if there is not. If you want to be semantically compatible with `NSInputStream`, return YES if a read must be attempted to determine if bytes are available.

## Tasks

### Creating Streams

+ [initWithData:](#) (page 6)

Creates and returns an initialized `NSInputStream` object for reading from a given `NSData` object.

+ [initWithFileAtPath:](#) (page 7)

Creates and returns an initialized `NSInputStream` object that reads data from the file at a given path.

- [initWithData:](#) (page 8)

Initializes and returns an `NSInputStream` object for reading from a given `NSData` object.

- [initWithFileAtPath:](#) (page 9)

Initializes and returns an `NSInputStream` object that reads data from the file at a given path.

### Using Streams

- [read:maxLength:](#) (page 9)

Reads up to a given number of bytes into a given buffer, and returns the actual number of bytes read.

- [getBuffer:length:](#) (page 7)

Returns by reference a pointer to a read buffer and, by reference, the number of bytes available, and returns a Boolean value that indicates whether the buffer is available.

- [hasBytesAvailable](#) (page 8)

Returns a Boolean value that indicates whether the receiver has bytes available to read.

## Class Methods

### **initWithData:**

Creates and returns an initialized `NSInputStream` object for reading from a given `NSData` object.

```
+ (id)initWithData:(NSData *)data
```

**Parameters***data*

The data object from which to read. The contents of *data* are copied.

**Return Value**

An initialized `NSInputStream` object for reading from *data*. If *data* is not an `NSData` object, this method returns `nil`.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

+ [inputStreamWithFileAtPath:](#) (page 7)

- [initWithData:](#) (page 8)

**Declared In**

`NSStream.h`

**inputStreamWithFileAtPath:**

Creates and returns an initialized `NSInputStream` object that reads data from the file at a given path.

```
+ (id)inputStreamWithFileAtPath:(NSString *)path
```

**Parameters***path*

The path to the file.

**Return Value**

An initialized `NSInputStream` object that reads data from the file at *path*. If the file specified by *path* doesn't exist or is unreadable, returns `nil`.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

+ [inputStreamWithData:](#) (page 6)

- [initWithFileAtPath:](#) (page 9)

**Declared In**

`NSStream.h`

## Instance Methods

**getBuffer:length:**

Returns by reference a pointer to a read buffer and, by reference, the number of bytes available, and returns a Boolean value that indicates whether the buffer is available.

```
- (BOOL)getBuffer:(uint8_t **)buffer length:(NSUInteger *)len
```

## Parameters

*buffer*

Upon return, contains a pointer to a read buffer. The buffer is only valid until the next stream operation is performed.

*len*

Upon return, contains the number of bytes available.

## Return Value

YES if the buffer is available, otherwise NO.

Subclasses of `NSInputStream` may return NO if this operation is not appropriate for the stream type.

## Availability

Available in Mac OS X v10.3 and later.

## Declared In

`NSStream.h`

## hasBytesAvailable

Returns a Boolean value that indicates whether the receiver has bytes available to read.

- (BOOL)hasBytesAvailable

## Return Value

YES if the receiver has bytes available to read, otherwise NO. May also return YES if a read must be attempted in order to determine the availability of bytes.

## Availability

Available in Mac OS X v10.3 and later.

## Declared In

`NSStream.h`

## initWithData:

Initializes and returns an `NSInputStream` object for reading from a given `NSData` object.

- (id)initWithData:(NSData \*)data

## Parameters

*data*

The data object from which to read. The contents of *data* are copied.

## Return Value

An initialized `NSInputStream` object for reading from *data*.

## Availability

Available in Mac OS X v10.3 and later.

## See Also

- [initWithFileAtPath:](#) (page 9)

+ [inputStreamWithData:](#) (page 6)



## Declared In

NSStream.h

## initWithFileAtPath:

Initializes and returns an `NSInputStream` object that reads data from the file at a given path.

```
- (id)initWithFileAtPath:(NSString *)path
```

### Parameters

*path*

The path to the file.

### Return Value

An initialized `NSInputStream` object that reads data from the file at *path*. If the file specified by *path* doesn't exist or is unreadable, returns `nil`.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [initWithData:](#) (page 8)

+ [inputStreamWithFileAtPath:](#) (page 7)

## Declared In

NSStream.h

## read:maxLength:

Reads up to a given number of bytes into a given buffer, and returns the actual number of bytes read.

```
- (NSInteger)read:(uint8_t *)buffer maxLength:(NSUInteger)len
```

### Parameters

*buffer*

A data buffer. The buffer must be large enough to contain the number of bytes specified by *len*.

*len*

The maximum number of bytes to read.

### Return Value

The actual number of bytes read.

### Availability

Available in Mac OS X v10.3 and later.

### Related Sample Code

CocoaEcho

CocoaHTTPServer

CocoaSOAP

## Declared In

NSStream.h



# Document Revision History

---

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

Date	Notes
2007-04-05	Made editorial improvements.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## G

---

getBuffer:length: **instance method** [7](#)

## H

---

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

## I

---

initWithData: **instance method** [8](#)

initWithFileAtPath: **instance method** [9](#)

initWithStreamWithData: **class method** [6](#)

initWithStreamWithFileAtPath: **class method** [7](#)

## R

---

read:maxLength: **instance method** [9](#)