| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DataInput.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/Console.html)   [**NEXT CLASS**](http://docs.google.com/java/io/DataInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/DataInput.html)    [**NO FRAMES**](http://docs.google.com/DataInput.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.io**

Interface DataInput

**All Known Subinterfaces:** [ImageInputStream](http://docs.google.com/javax/imageio/stream/ImageInputStream.html), [ImageOutputStream](http://docs.google.com/javax/imageio/stream/ImageOutputStream.html), [ObjectInput](http://docs.google.com/java/io/ObjectInput.html) **All Known Implementing Classes:** [DataInputStream](http://docs.google.com/java/io/DataInputStream.html), [FileCacheImageInputStream](http://docs.google.com/javax/imageio/stream/FileCacheImageInputStream.html), [FileCacheImageOutputStream](http://docs.google.com/javax/imageio/stream/FileCacheImageOutputStream.html), [FileImageInputStream](http://docs.google.com/javax/imageio/stream/FileImageInputStream.html), [FileImageOutputStream](http://docs.google.com/javax/imageio/stream/FileImageOutputStream.html), [ImageInputStreamImpl](http://docs.google.com/javax/imageio/stream/ImageInputStreamImpl.html), [ImageOutputStreamImpl](http://docs.google.com/javax/imageio/stream/ImageOutputStreamImpl.html), [MemoryCacheImageInputStream](http://docs.google.com/javax/imageio/stream/MemoryCacheImageInputStream.html), [MemoryCacheImageOutputStream](http://docs.google.com/javax/imageio/stream/MemoryCacheImageOutputStream.html), [ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html), [RandomAccessFile](http://docs.google.com/java/io/RandomAccessFile.html)

public interface **DataInput**

The DataInput interface provides for reading bytes from a binary stream and reconstructing from them data in any of the Java primitive types. There is also a facility for reconstructing a String from data in [modified UTF-8](#_3znysh7) format.

It is generally true of all the reading routines in this interface that if end of file is reached before the desired number of bytes has been read, an EOFException (which is a kind of IOException) is thrown. If any byte cannot be read for any reason other than end of file, an IOException other than EOFException is thrown. In particular, an IOException may be thrown if the input stream has been closed.

#### Modified UTF-8

Implementations of the DataInput and DataOutput interfaces represent Unicode strings in a format that is a slight modification of UTF-8. (For information regarding the standard UTF-8 format, see section *3.9 Unicode Encoding Forms* of *The Unicode Standard, Version 4.0*). Note that in the following tables, the most significant bit appears in the far left-hand column.

All characters in the range '\u0001' to '\u007F' are represented by a single byte:

|  | Bit Values |
| --- | --- |
| Byte 1 | | 0 | bits 6-0 | | | | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |

The null character '\u0000' and characters in the range '\u0080' to '\u07FF' are represented by a pair of bytes:

|  | Bit Values |
| --- | --- |
| Byte 1 | | 1 | 1 | 0 | bits 10-6 | | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |
| Byte 2 | | 1 | 0 | bits 5-0 | | | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |

char values in the range '\u0800' to '\uFFFF' are represented by three bytes:

|  | Bit Values |
| --- | --- |
| Byte 1 | | 1 | 1 | 1 | 0 | bits 15-12 | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |
| Byte 2 | | 1 | 0 | bits 11-6 | | | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |
| Byte 3 | | 1 | 0 | bits 5-0 | | | | | | | --- | --- | --- | --- | --- | --- | --- | --- | |

The differences between this format and the standard UTF-8 format are the following:

* The null byte '\u0000' is encoded in 2-byte format rather than 1-byte, so that the encoded strings never have embedded nulls.
* Only the 1-byte, 2-byte, and 3-byte formats are used.
* [Supplementary characters](http://docs.google.com/lang/Character.html#unicode) are represented in the form of surrogate pairs.

**Since:** JDK1.0 **See Also:**[DataInputStream](http://docs.google.com/java/io/DataInputStream.html), [DataOutput](http://docs.google.com/java/io/DataOutput.html)

| **Method Summary** | |
| --- | --- |
| boolean | [**readBoolean**](http://docs.google.com/java/io/DataInput.html#readBoolean())()            Reads one input byte and returns true if that byte is nonzero, false if that byte is zero. |
| byte | [**readByte**](http://docs.google.com/java/io/DataInput.html#readByte())()            Reads and returns one input byte. |
| char | [**readChar**](http://docs.google.com/java/io/DataInput.html#readChar())()            Reads two input bytes and returns a char value. |
| double | [**readDouble**](http://docs.google.com/java/io/DataInput.html#readDouble())()            Reads eight input bytes and returns a double value. |
| float | [**readFloat**](http://docs.google.com/java/io/DataInput.html#readFloat())()            Reads four input bytes and returns a float value. |
| void | [**readFully**](http://docs.google.com/java/io/DataInput.html#readFully(byte%5B%5D))(byte[] b)            Reads some bytes from an input stream and stores them into the buffer array b. |
| void | [**readFully**](http://docs.google.com/java/io/DataInput.html#readFully(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Reads len bytes from an input stream. |
| int | [**readInt**](http://docs.google.com/java/io/DataInput.html#readInt())()            Reads four input bytes and returns an int value. |
| [String](http://docs.google.com/java/lang/String.html) | [**readLine**](http://docs.google.com/java/io/DataInput.html#readLine())()            Reads the next line of text from the input stream. |
| long | [**readLong**](http://docs.google.com/java/io/DataInput.html#readLong())()            Reads eight input bytes and returns a long value. |
| short | [**readShort**](http://docs.google.com/java/io/DataInput.html#readShort())()            Reads two input bytes and returns a short value. |
| int | [**readUnsignedByte**](http://docs.google.com/java/io/DataInput.html#readUnsignedByte())()            Reads one input byte, zero-extends it to type int, and returns the result, which is therefore in the range 0 through 255. |
| int | [**readUnsignedShort**](http://docs.google.com/java/io/DataInput.html#readUnsignedShort())()            Reads two input bytes and returns an int value in the range 0 through 65535. |
| [String](http://docs.google.com/java/lang/String.html) | [**readUTF**](http://docs.google.com/java/io/DataInput.html#readUTF())()            Reads in a string that has been encoded using a [modified UTF-8](#_3znysh7) format. |
| int | [**skipBytes**](http://docs.google.com/java/io/DataInput.html#skipBytes(int))(int n)            Makes an attempt to skip over n bytes of data from the input stream, discarding the skipped bytes. |

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

### readFully

void **readFully**(byte[] b)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads some bytes from an input stream and stores them into the buffer array b. The number of bytes read is equal to the length of b.

This method blocks until one of the following conditions occurs:

* b.length bytes of input data are available, in which case a normal return is made.
* End of file is detected, in which case an EOFException is thrown.
* An I/O error occurs, in which case an IOException other than EOFException is thrown.

If b is null, a NullPointerException is thrown. If b.length is zero, then no bytes are read. Otherwise, the first byte read is stored into element b[0], the next one into b[1], and so on. If an exception is thrown from this method, then it may be that some but not all bytes of b have been updated with data from the input stream.

**Parameters:**b - the buffer into which the data is read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readFully

void **readFully**(byte[] b,  
 int off,  
 int len)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads len bytes from an input stream.

This method blocks until one of the following conditions occurs:

* len bytes of input data are available, in which case a normal return is made.
* End of file is detected, in which case an EOFException is thrown.
* An I/O error occurs, in which case an IOException other than EOFException is thrown.

If b is null, a NullPointerException is thrown. If off is negative, or len is negative, or off+len is greater than the length of the array b, then an IndexOutOfBoundsException is thrown. If len is zero, then no bytes are read. Otherwise, the first byte read is stored into element b[off], the next one into b[off+1], and so on. The number of bytes read is, at most, equal to len.

**Parameters:**b - the buffer into which the data is read.off - an int specifying the offset into the data.len - an int specifying the number of bytes to read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### skipBytes

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

Makes an attempt to skip over n bytes of data from the input stream, discarding the skipped bytes. However, it may skip over some smaller number of bytes, possibly zero. This may result from any of a number of conditions; reaching end of file before n bytes have been skipped is only one possibility. This method never throws an EOFException. The actual number of bytes skipped is returned.

**Parameters:**n - the number of bytes to be skipped. **Returns:**the number of bytes actually skipped. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readBoolean

boolean **readBoolean**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads one input byte and returns true if that byte is nonzero, false if that byte is zero. This method is suitable for reading the byte written by the writeBoolean method of interface DataOutput.

**Returns:**the boolean value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readByte

byte **readByte**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads and returns one input byte. The byte is treated as a signed value in the range -128 through 127, inclusive. This method is suitable for reading the byte written by the writeByte method of interface DataOutput.

**Returns:**the 8-bit value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readUnsignedByte

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

Reads one input byte, zero-extends it to type int, and returns the result, which is therefore in the range 0 through 255. This method is suitable for reading the byte written by the writeByte method of interface DataOutput if the argument to writeByte was intended to be a value in the range 0 through 255.

**Returns:**the unsigned 8-bit value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readShort

short **readShort**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads two input bytes and returns a short value. Let a be the first byte read and b be the second byte. The value returned is:

(short)((a << 8) | (b & 0xff))

This method is suitable for reading the bytes written by the writeShort method of interface DataOutput.

**Returns:**the 16-bit value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readUnsignedShort

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

Reads two input bytes and returns an int value in the range 0 through 65535. Let a be the first byte read and b be the second byte. The value returned is:

(((a & 0xff) << 8) | (b & 0xff))

This method is suitable for reading the bytes written by the writeShort method of interface DataOutput if the argument to writeShort was intended to be a value in the range 0 through 65535.

**Returns:**the unsigned 16-bit value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readChar

char **readChar**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads two input bytes and returns a char value. Let a be the first byte read and b be the second byte. The value returned is:

(char)((a << 8) | (b & 0xff))

This method is suitable for reading bytes written by the writeChar method of interface DataOutput.

**Returns:**the char value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readInt

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

Reads four input bytes and returns an int value. Let a-d be the first through fourth bytes read. The value returned is:

(((a & 0xff) << 24) | ((b & 0xff) << 16) |  
 ((c & 0xff) << 8) | (d & 0xff))

This method is suitable for reading bytes written by the writeInt method of interface DataOutput.

**Returns:**the int value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readLong

long **readLong**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads eight input bytes and returns a long value. Let a-h be the first through eighth bytes read. The value returned is:

(((long)(a & 0xff) << 56) |  
 ((long)(b & 0xff) << 48) |  
 ((long)(c & 0xff) << 40) |  
 ((long)(d & 0xff) << 32) |  
 ((long)(e & 0xff) << 24) |  
 ((long)(f & 0xff) << 16) |  
 ((long)(g & 0xff) << 8) |  
 ((long)(h & 0xff)))

This method is suitable for reading bytes written by the writeLong method of interface DataOutput.

**Returns:**the long value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readFloat

float **readFloat**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads four input bytes and returns a float value. It does this by first constructing an int value in exactly the manner of the readInt method, then converting this int value to a float in exactly the manner of the method Float.intBitsToFloat. This method is suitable for reading bytes written by the writeFloat method of interface DataOutput.

**Returns:**the float value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readDouble

double **readDouble**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads eight input bytes and returns a double value. It does this by first constructing a long value in exactly the manner of the readlong method, then converting this long value to a double in exactly the manner of the method Double.longBitsToDouble. This method is suitable for reading bytes written by the writeDouble method of interface DataOutput.

**Returns:**the double value read. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readLine

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

Reads the next line of text from the input stream. It reads successive bytes, converting each byte separately into a character, until it encounters a line terminator or end of file; the characters read are then returned as a String. Note that because this method processes bytes, it does not support input of the full Unicode character set.

If end of file is encountered before even one byte can be read, then null is returned. Otherwise, each byte that is read is converted to type char by zero-extension. If the character '\n' is encountered, it is discarded and reading ceases. If the character '\r' is encountered, it is discarded and, if the following byte converts to the character '\n', then that is discarded also; reading then ceases. If end of file is encountered before either of the characters '\n' and '\r' is encountered, reading ceases. Once reading has ceased, a String is returned that contains all the characters read and not discarded, taken in order. Note that every character in this string will have a value less than \u0100, that is, (char)256.

**Returns:**the next line of text from the input stream, or null if the end of file is encountered before a byte can be read. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs.

### readUTF

[String](http://docs.google.com/java/lang/String.html) **readUTF**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads in a string that has been encoded using a [modified UTF-8](#_3znysh7) format. The general contract of readUTF is that it reads a representation of a Unicode character string encoded in modified UTF-8 format; this string of characters is then returned as a String.

First, two bytes are read and used to construct an unsigned 16-bit integer in exactly the manner of the readUnsignedShort method . This integer value is called the *UTF length* and specifies the number of additional bytes to be read. These bytes are then converted to characters by considering them in groups. The length of each group is computed from the value of the first byte of the group. The byte following a group, if any, is the first byte of the next group.

If the first byte of a group matches the bit pattern 0xxxxxxx (where x means "may be 0 or 1"), then the group consists of just that byte. The byte is zero-extended to form a character.

If the first byte of a group matches the bit pattern 110xxxxx, then the group consists of that byte a and a second byte b. If there is no byte b (because byte a was the last of the bytes to be read), or if byte b does not match the bit pattern 10xxxxxx, then a UTFDataFormatException is thrown. Otherwise, the group is converted to the character:

(char)(((a& 0x1F) << 6) | (b & 0x3F))

If the first byte of a group matches the bit pattern 1110xxxx, then the group consists of that byte a and two more bytes b and c. If there is no byte c (because byte a was one of the last two of the bytes to be read), or either byte b or byte c does not match the bit pattern 10xxxxxx, then a UTFDataFormatException is thrown. Otherwise, the group is converted to the character:

(char)(((a & 0x0F) << 12) | ((b & 0x3F) << 6) | (c & 0x3F))

If the first byte of a group matches the pattern 1111xxxx or the pattern 10xxxxxx, then a UTFDataFormatException is thrown.

If end of file is encountered at any time during this entire process, then an EOFException is thrown.

After every group has been converted to a character by this process, the characters are gathered, in the same order in which their corresponding groups were read from the input stream, to form a String, which is returned.

The writeUTF method of interface DataOutput may be used to write data that is suitable for reading by this method.

**Returns:**a Unicode string. **Throws:** [EOFException](http://docs.google.com/java/io/EOFException.html) - if this stream reaches the end before reading all the bytes. [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs. [UTFDataFormatException](http://docs.google.com/java/io/UTFDataFormatException.html) - if the bytes do not represent a valid modified UTF-8 encoding of a string.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DataInput.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/Console.html)   [**NEXT CLASS**](http://docs.google.com/java/io/DataInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/io/DataInput.html)    [**NO FRAMES**](http://docs.google.com/DataInput.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#2et92p0) | 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).