

Package java.io

Provides for system input and output through data streams, serialization and the file system.

See: [Description](#)

Interface Summary	
Interface	Description
Closeable	A Closeable is a source or destination of data that can be closed.
DataInput	The DataInput interface provides for reading bytes from a binary stream and reconstructing from them data in any of the Java primitive types.
DataOutput	The DataOutput interface provides for converting data from any of the Java primitive types to a series of bytes and writing these bytes to a binary stream.
Externalizable	Only the identity of the class of an Externalizable instance is written in the serialization stream and it is the responsibility of the class to save and restore the contents of its instances.
FileFilter	A filter for abstract pathnames.
FilenameFilter	Instances of classes that implement this interface are used to filter filenames.
Flushable	A Flushable is a destination of data that can be flushed.
ObjectInput	ObjectInput extends the DataInput interface to include the reading of objects.
ObjectInputValidation	Callback interface to allow validation of objects within a graph.
ObjectOutput	ObjectOutput extends the DataOutput interface to include writing of objects.
ObjectStreamConstants	Constants written into the Object Serialization Stream.

Serializable	Serializability of a class is enabled by the class implementing the java.io.Serializable interface.
---------------------	---

Class Summary

Class	Description
BufferedInputStream	A BufferedInputStream adds functionality to another input stream-namely, the ability to buffer the input and to support the mark and reset methods.
BufferedOutputStream	The class implements a buffered output stream.
BufferedReader	Reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.
BufferedWriter	Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.
ByteArrayInputStream	A ByteArrayInputStream contains an internal buffer that contains bytes that may be read from the stream.
ByteArrayOutputStream	This class implements an output stream in which the data is written into a byte array.
CharArrayReader	This class implements a character buffer that can be used as a character-input stream.
CharArrayWriter	This class implements a character buffer that can be used as an Writer.
Console	Methods to access the character-based console device, if any, associated with the current Java virtual machine.
DataInputStream	A data input stream lets an application read primitive Java data types from an underlying input stream in a machine-independent way.
DataOutputStream	A data output stream lets an application write primitive Java data types to an output stream in a portable way.
File	An abstract representation of file and directory pathnames.
FileDescriptor	Instances of the file descriptor class serve as

	an opaque handle to the underlying machine-specific structure representing an open file, an open socket, or another source or sink of bytes.
FileInputStream	A <code>FileInputStream</code> obtains input bytes from a file in a file system.
FileOutputStream	A file output stream is an output stream for writing data to a <code>File</code> or to a <code>FileDescriptor</code> .
FilePermission	This class represents access to a file or directory.
FileReader	Convenience class for reading character files.
FileWriter	Convenience class for writing character files.
FilterInputStream	A <code>FilterInputStream</code> contains some other input stream, which it uses as its basic source of data, possibly transforming the data along the way or providing additional functionality.
FilterOutputStream	This class is the superclass of all classes that filter output streams.
FilterReader	Abstract class for reading filtered character streams.
FilterWriter	Abstract class for writing filtered character streams.
InputStream	This abstract class is the superclass of all classes representing an input stream of bytes.
InputStreamReader	An <code>InputStreamReader</code> is a bridge from byte streams to character streams: It reads bytes and decodes them into characters using a specified charset .
LineNumberInputStream	Deprecated <i>This class incorrectly assumes that bytes adequately represent characters.</i>
LineNumberReader	A buffered character-input stream that keeps track of line numbers.
ObjectInputStream	An <code>ObjectInputStream</code> deserializes primitive data and objects previously written using an <code>ObjectOutputStream</code> .
ObjectInputStream.GetField	Provide access to the persistent fields read from the input stream.
ObjectOutputStream	An <code>ObjectOutputStream</code> writes primitive data

	types and graphs of Java objects to an <code>OutputStream</code> .
<code>ObjectOutputStream.PutField</code>	Provide programmatic access to the persistent fields to be written to <code>ObjectOutput</code> .
<code>ObjectStreamClass</code>	Serialization's descriptor for classes.
<code>ObjectStreamField</code>	A description of a <code>Serializable</code> field from a <code>Serializable</code> class.
<code>OutputStream</code>	This abstract class is the superclass of all classes representing an output stream of bytes.
<code>OutputStreamWriter</code>	An <code>OutputStreamWriter</code> is a bridge from character streams to byte streams: Characters written to it are encoded into bytes using a specified <code>charset</code> .
<code>PipedInputStream</code>	A piped input stream should be connected to a piped output stream; the piped input stream then provides whatever data bytes are written to the piped output stream.
<code>PipedOutputStream</code>	A piped output stream can be connected to a piped input stream to create a communications pipe.
<code>PipedReader</code>	Piped character-input streams.
<code>PipedWriter</code>	Piped character-output streams.
<code>PrintStream</code>	A <code>PrintStream</code> adds functionality to another output stream, namely the ability to print representations of various data values conveniently.
<code>PrintWriter</code>	Prints formatted representations of objects to a text-output stream.
<code>PushbackInputStream</code>	A <code>PushbackInputStream</code> adds functionality to another input stream, namely the ability to "push back" or "unread" one byte.
<code>PushbackReader</code>	A character-stream reader that allows characters to be pushed back into the stream.
<code>RandomAccessFile</code>	Instances of this class support both reading and writing to a random access file.
<code>Reader</code>	Abstract class for reading character streams.
<code>SequenceInputStream</code>	A <code>SequenceInputStream</code> represents the logical concatenation of other input streams.
<code>SerializablePermission</code>	This class is for <code>Serializable</code> permissions.

StreamTokenizer	The StreamTokenizer class takes an input stream and parses it into "tokens", allowing the tokens to be read one at a time.
StringBufferInputStream	Deprecated <i>This class does not properly convert characters into bytes.</i>
StringReader	A character stream whose source is a string.
StringWriter	A character stream that collects its output in a string buffer, which can then be used to construct a string.
Writer	Abstract class for writing to character streams.

Exception Summary

Exception	Description
CharConversionException	Base class for character conversion exceptions.
EOFException	Signals that an end of file or end of stream has been reached unexpectedly during input.
FileNotFoundException	Signals that an attempt to open the file denoted by a specified pathname has failed.
InterruptedIOException	Signals that an I/O operation has been interrupted.
InvalidClassException	Thrown when the Serialization runtime detects one of the following problems with a Class.
InvalidObjectException	Indicates that one or more deserialized objects failed validation tests.
IOException	Signals that an I/O exception of some sort has occurred.
NotActiveException	Thrown when serialization or deserialization is not active.
NotSerializableException	Thrown when an instance is required to have a Serializable interface.
ObjectStreamException	Superclass of all exceptions specific to Object Stream classes.
OptionalDataException	Exception indicating the failure of an object

	read operation due to unread primitive data, or the end of data belonging to a serialized object in the stream.
StreamCorruptedException	Thrown when control information that was read from an object stream violates internal consistency checks.
SyncFailedException	Signals that a sync operation has failed.
UnsupportedEncodingException	The Character Encoding is not supported.
UTFDataFormatException	Signals that a malformed string in modified UTF-8 format has been read in a data input stream or by any class that implements the data input interface.
WriteAbortedException	Signals that one of the <code>ObjectStreamExceptions</code> was thrown during a write operation.

Error Summary

Error	Description
IOException	Thrown when a serious I/O error has occurred.

Package java.io Description

Provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a `NullPointerException` to be thrown.

Package Specification

- [Java Object Serialization Specification](#)

Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- [Serialization Enhancements](#)

Since:
JDK1.0