Java IO is an API that comes with Java which is targeted at reading and writing data (input and output). Most applications need to process some input and produce some output based on that input. For instance, read data from a file or over network, and write to a file or write a response back over the network.

The Java IO API is located in the Java IO package (java.io). If you look at the Java IO classes in the java.io package the vast amount of choices can be rather confusing. What is the purpose of all these classes? Which one should you choose for a given task? How do you create your own classes to plugin? etc

The purpose of this tutorial is to try to give you an overview of how all these classes are grouped, and the purpose behind them, so you don't have to wonder whether you chose the right class, or whether a class already exists for your purpose.

**Java IO Across Java Versions**

The Java IO API has remained reasonably stable across the many Java versions that has come since it was first created. However, small things have changed, like how to close an InputStream or OutputStream best using the try-with-resources constructs. This Java IO tutorial has been updated in many places to reflect these changes whenever they have occurred.

## The Scope of the Java IO (java.io) Package

The java.io package doesn't actually address all types of input and output. For instance, input from and output to a GUI or web page is not covered in the Java IO package. Those types of input are covered elsewhere, for instance by the JFC classes in the Swing project, or the [**Servlet**](http://tutorials.jenkov.com/java-servlets/index.html) and HTTP packages in the Java Enterprise Edition.

The Java IO package is primarily focused on input and output to files, network streams, internal memory buffers etc. However, the Java IO package does not contain classes to open network sockets which are necessary for network communication. For that purpose you need to use the [**Java Networking API**](http://tutorials.jenkov.com/java-networking/index.html). Once you have opened a socket (network connection) though, you read and write data to and from it via Java IO's InputStream and OutputStream classes.

## Java NIO - The Alternative IO API

Java also contains another IO API called [**Java NIO**](http://tutorials.jenkov.com/java-nio/index.html). It contains classes that does much of the same as the Java IO and Java Networking APIs, but Java NIO can work in non-blocking mode. Non-blocking IO can in some situations give a big performance boost over blocking IO.

**Java IO Class Overview Table**

Here is a table listing most (if not all) Java IO classes divided by input, output, being byte based or character based, and any more specific purpose they may be addressing, like buffering, parsing etc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Byte Based | | Character Based | |
|  | Input | Output | Input | Output |
| Basic | [**InputStream**](http://tutorials.jenkov.com/java-io/inputstream.html) | [**OutputStream**](http://tutorials.jenkov.com/java-io/outputstream.html) | [**Reader**](http://tutorials.jenkov.com/java-io/reader.html) [**InputStreamReader**](http://tutorials.jenkov.com/java-io/inputstreamreader.html) | [**Writer**](http://tutorials.jenkov.com/java-io/writer.hml) [**OutputStreamWriter**](http://tutorials.jenkov.com/java-io/outputstreamwriter.html) |
| Arrays | [**ByteArrayInputStream**](http://tutorials.jenkov.com/java-io/bytearrayinputstream.html) | [**ByteArrayOutputStream**](http://tutorials.jenkov.com/java-io/bytearrayoutputstream.html) | [**CharArrayReader**](http://tutorials.jenkov.com/java-io/chararrayreader.html) | [**CharArrayWriter**](http://tutorials.jenkov.com/java-io/chararraywriter.html) |
| Files | [**FileInputStream**](http://tutorials.jenkov.com/java-io/fileinputstream.html) [**RandomAccessFile**](http://tutorials.jenkov.com/java-io/randomaccessfile.html) | [**FileOutputStream**](http://tutorials.jenkov.com/java-io/fileoutputstream.html) [**RandomAccessFile**](http://tutorials.jenkov.com/java-io/randomaccessfile.html) | [**FileReader**](http://tutorials.jenkov.com/java-io/filereader.html) | [**FileWriter**](http://tutorials.jenkov.com/java-io/filewriter.html) |
| Pipes | [**PipedInputStream**](http://tutorials.jenkov.com/java-io/pipedinputstream.html) | [**PipedOutputStream**](http://tutorials.jenkov.com/java-io/pipedoutputstream.html) | [**PipedReader**](http://tutorials.jenkov.com/java-io/pipedreader.html) | [**PipedWriter**](http://tutorials.jenkov.com/java-io/pipedwriter.html) |
| Buffering | [**BufferedInputStream**](http://tutorials.jenkov.com/java-io/bufferedinputstream.html) | [**BufferedOutputStream**](http://tutorials.jenkov.com/java-io/bufferedoutputstream.html) | [**BufferedReader**](http://tutorials.jenkov.com/java-io/bufferedreader.html) | [**BufferedWriter**](http://tutorials.jenkov.com/java-io/bufferedwriter.html) |
| Filtering | [**FilterInputStream**](http://tutorials.jenkov.com/java-io/filterinputstream.html) | [**FilterOutputStream**](http://tutorials.jenkov.com/java-io/filteroutputstream.html) | [**FilterReader**](http://tutorials.jenkov.com/java-io/filterreader.html) | [**FilterWriter**](http://tutorials.jenkov.com/java-io/filterwriter.html) |
| Parsing | [**PushbackInputStream**](http://tutorials.jenkov.com/java-io/pushbackinputstream.html) [**StreamTokenizer**](http://tutorials.jenkov.com/java-io/streamtokenizer.html) |  | [**PushbackReader**](http://tutorials.jenkov.com/java-io/pushbackreader.html) [**LineNumberReader**](http://tutorials.jenkov.com/java-io/linenumberreader.html) |  |
| Strings |  |  | [**StringReader**](http://tutorials.jenkov.com/java-io/stringreader.html) | [**StringWriter**](http://tutorials.jenkov.com/java-io/stringwriter.html) |
| Data | [**DataInputStream**](http://tutorials.jenkov.com/java-io/datainputstream.html) | [**DataOutputStream**](http://tutorials.jenkov.com/java-io/dataoutputstream.html) |  |  |
| Data - Formatted |  | [**PrintStream**](http://tutorials.jenkov.com/java-io/printstream.html) |  | [**PrintWriter**](http://tutorials.jenkov.com/java-io/printwriter.html) |
| Objects | [**ObjectInputStream**](http://tutorials.jenkov.com/java-io/objectinputstream.html) | [**ObjectOutputStream**](http://tutorials.jenkov.com/java-io/objectoutputstream.html) |  |  |
| Utilities | [**SequenceInputStream**](http://tutorials.jenkov.com/java-io/sequenceinputstream.html) |  |  |  |

* [Input and Output - Source and Destination](http://tutorials.jenkov.com/java-io/overview.html#inputoutput)
  + [Streams](http://tutorials.jenkov.com/java-io/overview.html#streams)
  + [The InputStream, OutputStream, Reader and Writer](http://tutorials.jenkov.com/java-io/overview.html#inputstream-outputstream-reader-writer)
* [Java IO Purposes and Features](http://tutorials.jenkov.com/java-io/overview.html#purposes)
* [Java IO Class Overview Table](http://tutorials.jenkov.com/java-io/overview.html#overview)

In this text I will try to give you an overview of the classes in the Java IO (java.io) package. More specifically, I will try to group the classes after their purpose. This grouping should make it easier for you in the future, to determine the purpose of a class, or find the class you need for a specific purpose.

## Input and Output - Source and Destination

The terms "input" and "output" can sometimes be a bit confusing. The input of one part of an application is often the output of another. Is an OutputStream a stream where output is written to, or output comes out from (for you to read)? After all, an InputStream outputs its data to the reading program, doesn't it? Personally, I found this a bit confusing back in the day when I first started out learning about Java IO.

In an attempt to clear out this possible confusion, I have tried to put some different names on input and output to try to link them conceptually to where the input comes from, and where the output goes.

Java's IO package mostly concerns itself with the reading of raw data from a source and writing of raw data to a destination. The most typical sources and destinations of data are these:

* Files
* Pipes
* Network Connections
* In-memory Buffers (e.g. arrays)
* System.in, System.out, System.error

The diagram below illustrates the principle of a program reading data from a source and writing it to some destination:

SourceProgramDestination

### Streams

[**IO Streams**](http://tutorials.jenkov.com/java-io/streams.html) are a core concept in Java IO. A stream is a conceptually endless flow of data. You can either read from a stream or write to a stream. A stream is connected to a data source or a data destination. Streams in Java IO can be either byte based (reading and writing bytes) or character based (reading and writing characters).

### The InputStream, OutputStream, Reader and Writer

A program that needs to read data from some source needs an **[InputStream](http://tutorials.jenkov.com/java-io/inputstream.html)** or a [**Reader**](http://tutorials.jenkov.com/java-io/reader.html). A program that needs to write data to some destination needs an **[OutputStream](http://tutorials.jenkov.com/java-io/outputstream.html)** or a [**Writer**](http://tutorials.jenkov.com/java-io/writer.html). This is also illustrated in the diagram below:

SourceInputStream /ReaderProgramProgramOutputStream /WriterDestination

An InputStream or Reader is linked to a **source** of data. An OutputStream or Writer is linked to a **destination** of data.

## Java IO Purposes and Features

Java IO contains many subclasses of the InputStream, OutputStream, Reader and Writer classes. The reason is, that all of these subclasses are addressing various different purposes. That is why there are so many different classes. The purposes addressed are summarized below:

* File Access
* Network Access
* Internal Memory Buffer Access
* Inter-Thread Communication (Pipes)
* Buffering
* Filtering
* Parsing
* Reading and Writing Text (Readers / Writers)
* Reading and Writing Primitive Data (long, int etc.)
* Reading and Writing Objects

These purposes are nice to know about when reading through the Java IO classes. They make it somewhat easier to understand what the classes are targeting.

## Java IO Class Overview Table

Having discussed sources, destinations, input, output and the various IO purposes targeted by the Java IO classes, here is a table listing most (if not all) Java IO classes divided by input, output, being byte based or character based, and any more specific purpose they may be addressing, like buffering, parsing etc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Byte Based | | Character Based | |
|  | Input | Output | Input | Output |
| Basic | [**InputStream**](http://tutorials.jenkov.com/java-io/inputstream.html) | [**OutputStream**](http://tutorials.jenkov.com/java-io/outputstream.html) | [**Reader**](http://tutorials.jenkov.com/java-io/reader.html) [**InputStreamReader**](http://tutorials.jenkov.com/java-io/inputstreamreader.html) | [**Writer**](http://tutorials.jenkov.com/java-io/writer.hml) [**OutputStreamWriter**](http://tutorials.jenkov.com/java-io/outputstreamwriter.html) |
| Arrays | [**ByteArrayInputStream**](http://tutorials.jenkov.com/java-io/bytearrayinputstream.html) | [**ByteArrayOutputStream**](http://tutorials.jenkov.com/java-io/bytearrayoutputstream.html) | [**CharArrayReader**](http://tutorials.jenkov.com/java-io/chararrayreader.html) | [**CharArrayWriter**](http://tutorials.jenkov.com/java-io/chararraywriter.html) |
| Files | [**FileInputStream**](http://tutorials.jenkov.com/java-io/fileinputstream.html) [**RandomAccessFile**](http://tutorials.jenkov.com/java-io/randomaccessfile.html) | [**FileOutputStream**](http://tutorials.jenkov.com/java-io/fileoutputstream.html) [**RandomAccessFile**](http://tutorials.jenkov.com/java-io/randomaccessfile.html) | [**FileReader**](http://tutorials.jenkov.com/java-io/filereader.html) | [**FileWriter**](http://tutorials.jenkov.com/java-io/filewriter.html) |
| Pipes | [**PipedInputStream**](http://tutorials.jenkov.com/java-io/pipedinputstream.html) | [**PipedOutputStream**](http://tutorials.jenkov.com/java-io/pipedoutputstream.html) | [**PipedReader**](http://tutorials.jenkov.com/java-io/pipedreader.html) | [**PipedWriter**](http://tutorials.jenkov.com/java-io/pipedwriter.html) |
| Buffering | [**BufferedInputStream**](http://tutorials.jenkov.com/java-io/bufferedinputstream.html) | [**BufferedOutputStream**](http://tutorials.jenkov.com/java-io/bufferedoutputstream.html) | [**BufferedReader**](http://tutorials.jenkov.com/java-io/bufferedreader.html) | [**BufferedWriter**](http://tutorials.jenkov.com/java-io/bufferedwriter.html) |
| Filtering | [**FilterInputStream**](http://tutorials.jenkov.com/java-io/filterinputstream.html) | [**FilterOutputStream**](http://tutorials.jenkov.com/java-io/filteroutputstream.html) | [**FilterReader**](http://tutorials.jenkov.com/java-io/filterreader.html) | [**FilterWriter**](http://tutorials.jenkov.com/java-io/filterwriter.html) |
| Parsing | [**PushbackInputStream**](http://tutorials.jenkov.com/java-io/pushbackinputstream.html) [**StreamTokenizer**](http://tutorials.jenkov.com/java-io/streamtokenizer.html) |  | [**PushbackReader**](http://tutorials.jenkov.com/java-io/pushbackreader.html) [**LineNumberReader**](http://tutorials.jenkov.com/java-io/linenumberreader.html) |  |
| Strings |  |  | [**StringReader**](http://tutorials.jenkov.com/java-io/stringreader.html) | [**StringWriter**](http://tutorials.jenkov.com/java-io/stringwriter.html) |
| Data | [**DataInputStream**](http://tutorials.jenkov.com/java-io/datainputstream.html) | [**DataOutputStream**](http://tutorials.jenkov.com/java-io/dataoutputstream.html) |  |  |
| Data - Formatted |  | [**PrintStream**](http://tutorials.jenkov.com/java-io/printstream.html) |  | [**PrintWriter**](http://tutorials.jenkov.com/java-io/printwriter.html) |
| Objects | [**ObjectInputStream**](http://tutorials.jenkov.com/java-io/objectinputstream.html) | [**ObjectOutputStream**](http://tutorials.jenkov.com/java-io/objectoutputstream.html) |  |  |
| Utilities | [**SequenceInputStream**](http://tutorials.jenkov.com/java-io/sequenceinputstream.html) |  |  |  |

* [Java IO File Classes](http://tutorials.jenkov.com/java-io/files.html#java-io-file-classes)
* [Reading Files via Java IO](http://tutorials.jenkov.com/java-io/files.html#reading)
* [Writing File via Java IO](http://tutorials.jenkov.com/java-io/files.html#writing)
* [Random Access to Files via Java IO](http://tutorials.jenkov.com/java-io/files.html#random)
* [File and Directory Info Access](http://tutorials.jenkov.com/java-io/files.html#directory)