Class java.lang.Process

```
java.lang.Object
   |
   +----java.lang.Process
```

public class **Process**
extends Object

An instance of class Process is returned by variants of the exec method in class System. From the Process instance, it is possible to: get the standin and/or standout of the subprocess, kill the subprocess, wait for it to terminate, and to retrieve the final exit value of the process.

Dropping the last reference to a Process instance does *not* kill the subprocess. There is no requirement that the subprocess execute asynchronously with the existing Java process.

Constructor Index

o **Process**()

Method Index

o **destroy**()
  Kills the subprocess.
o **exitValue**()
  Returns the exit value for the subprocess.
o **getErrorStream**()
  Returns the an InputStream connected to the error stream of the child process.
o **getInputStream**()
  Returns a Stream connected to the output of the child process.
o **getOutputStream**()
  Returns a Stream connected to the input of the child process.
o **waitFor**()
  Waits for the subprocess to complete.

Constructors

o Process

```
public Process()
```

Methods

o getOutputStream

```
public abstract OutputStream getOutputStream()
```

Returns a Stream connected to the input of the child process. This stream is traditionally buffered.

o getInputStream

```
public abstract InputStream getInputStream()
```

Returns a Stream connected to the output of the child process. This stream is traditionally buffered.

o getErrorStream

```
public abstract InputStream getErrorStream()
```

Returns the an InputStream connected to the error stream of the child process. This stream is traditionally unbuffered.

o waitFor

```
public abstract int waitFor() throws InterruptedException
```

Waits for the subprocess to complete. If the subprocess has already terminated the exit value is simply returned. If the subprocess has not yet terminated the calling thread will be blocked until the subprocess exits. Should this be called join()?

o exitValue

```
public abstract int exitValue()
```

Returns the exit value for the subprocess.
Throws: IllegalThreadStateException
If the subprocess has not yet terminated.

o destroy

```
public abstract void destroy()
```

Kills the subprocess.

All Packages This Package Previous Next