| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Process.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/lang/Package.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ProcessBuilder.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Process.html)    [**NO FRAMES**](http://docs.google.com/Process.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.lang**

Class Process

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.lang.Process**

public abstract class **Process**extends [Object](http://docs.google.com/java/lang/Object.html)

The [ProcessBuilder.start()](http://docs.google.com/java/lang/ProcessBuilder.html#start()) and [Runtime.exec](http://docs.google.com/java/lang/Runtime.html#exec(java.lang.String%5B%5D,%20java.lang.String%5B%5D,%20java.io.File)) methods create a native process and return an instance of a subclass of Process that can be used to control the process and obtain information about it. The class Process provides methods for performing input from the process, performing output to the process, waiting for the process to complete, checking the exit status of the process, and destroying (killing) the process.

The methods that create processes may not work well for special processes on certain native platforms, such as native windowing processes, daemon processes, Win16/DOS processes on Microsoft Windows, or shell scripts. The created subprocess does not have its own terminal or console. All its standard io (i.e. stdin, stdout, stderr) operations will be redirected to the parent process through three streams ([getOutputStream()](http://docs.google.com/java/lang/Process.html#getOutputStream()), [getInputStream()](http://docs.google.com/java/lang/Process.html#getInputStream()), [getErrorStream()](http://docs.google.com/java/lang/Process.html#getErrorStream())). The parent process uses these streams to feed input to and get output from the subprocess. Because some native platforms only provide limited buffer size for standard input and output streams, failure to promptly write the input stream or read the output stream of the subprocess may cause the subprocess to block, and even deadlock.

The subprocess is not killed when there are no more references to the Process object, but rather the subprocess continues executing asynchronously.

There is no requirement that a process represented by a Process object execute asynchronously or concurrently with respect to the Java process that owns the Process object.

**Since:** JDK1.0 **See Also:**[ProcessBuilder](http://docs.google.com/java/lang/ProcessBuilder.html), [Runtime.exec(String[], String[], File)](http://docs.google.com/java/lang/Runtime.html#exec(java.lang.String%5B%5D,%20java.lang.String%5B%5D,%20java.io.File))

| **Constructor Summary** | |
| --- | --- |
| [**Process**](http://docs.google.com/java/lang/Process.html#Process())() |

| **Method Summary** | |
| --- | --- |
| abstract  void | [**destroy**](http://docs.google.com/java/lang/Process.html#destroy())()            Kills the subprocess. |
| abstract  int | [**exitValue**](http://docs.google.com/java/lang/Process.html#exitValue())()            Returns the exit value for the subprocess. |
| abstract  [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getErrorStream**](http://docs.google.com/java/lang/Process.html#getErrorStream())()            Gets the error stream of the subprocess. |
| abstract  [InputStream](http://docs.google.com/java/io/InputStream.html) | [**getInputStream**](http://docs.google.com/java/lang/Process.html#getInputStream())()            Gets the input stream of the subprocess. |
| abstract  [OutputStream](http://docs.google.com/java/io/OutputStream.html) | [**getOutputStream**](http://docs.google.com/java/lang/Process.html#getOutputStream())()            Gets the output stream of the subprocess. |
| abstract  int | [**waitFor**](http://docs.google.com/java/lang/Process.html#waitFor())()            causes the current thread to wait, if necessary, until the process represented by this Process object has terminated. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### Process

public **Process**()

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

### getOutputStream

public abstract [OutputStream](http://docs.google.com/java/io/OutputStream.html) **getOutputStream**()

Gets the output stream of the subprocess. Output to the stream is piped into the standard input stream of the process represented by this Process object.

Implementation note: It is a good idea for the output stream to be buffered.

**Returns:**the output stream connected to the normal input of the subprocess.

### getInputStream

public abstract [InputStream](http://docs.google.com/java/io/InputStream.html) **getInputStream**()

Gets the input stream of the subprocess. The stream obtains data piped from the standard output stream of the process represented by this Process object.

Implementation note: It is a good idea for the input stream to be buffered.

**Returns:**the input stream connected to the normal output of the subprocess.**See Also:**[ProcessBuilder.redirectErrorStream()](http://docs.google.com/java/lang/ProcessBuilder.html#redirectErrorStream())

### getErrorStream

public abstract [InputStream](http://docs.google.com/java/io/InputStream.html) **getErrorStream**()

Gets the error stream of the subprocess. The stream obtains data piped from the error output stream of the process represented by this Process object.

Implementation note: It is a good idea for the input stream to be buffered.

**Returns:**the input stream connected to the error stream of the subprocess.**See Also:**[ProcessBuilder.redirectErrorStream()](http://docs.google.com/java/lang/ProcessBuilder.html#redirectErrorStream())

### waitFor

public abstract int **waitFor**()  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

causes the current thread to wait, if necessary, until the process represented by this Process object has terminated. This method returns immediately if the subprocess has already terminated. If the subprocess has not yet terminated, the calling thread will be blocked until the subprocess exits.

**Returns:**the exit value of the process. By convention, 0 indicates normal termination. **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if the current thread is [interrupted](http://docs.google.com/java/lang/Thread.html#interrupt()) by another thread while it is waiting, then the wait is ended and an [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) is thrown.

### exitValue

public abstract int **exitValue**()

Returns the exit value for the subprocess.

**Returns:**the exit value of the subprocess represented by this Process object. by convention, the value 0 indicates normal termination. **Throws:** [IllegalThreadStateException](http://docs.google.com/java/lang/IllegalThreadStateException.html) - if the subprocess represented by this Process object has not yet terminated.

### destroy

public abstract void **destroy**()

Kills the subprocess. The subprocess represented by this Process object is forcibly terminated.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Process.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/lang/Package.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ProcessBuilder.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/Process.html)    [**NO FRAMES**](http://docs.google.com/Process.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).