
NSProcessInfo

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSProcessInfo.h

Class Description

The `NSProcessInfo` class provides methods to access process-wide information. An `NSProcessInfo` object can return such information as the arguments, environment variables, host name, or process name. The **`processInfo`** class method returns a shared `NSProcessInfo` object for the process. For example, the following line returns the `NSProcessInfo` object, which then provides the name of the current process:

```
NSString *processName = [[NSProcessInfo processInfo] processName];
```

`NSProcessInfo` also includes the **`operatingSystem`** method, which returns an **enum** constant identifying the operating system on which the process is executing.

Method Types

Getting an <code>NSProcessInfo</code> object	+ <code>processInfo</code>
Returning process information	- <code>arguments</code> - <code>environment</code> - <code>hostName</code> - <code>processName</code> - <code>globallyUniqueString</code>
Returning the host operating system	- <code>operatingSystem</code>
Specifying a process name	- <code>setProcessName:</code>

Class Methods

`processInfo`

+ (`NSProcessInfo` *)**`processInfo`**

Returns an initialized `NSProcessInfo` object for the process. An `NSProcessInfo` object is created the first time this method is invoked, and that same object is returned on each subsequent invocation.

Instance Methods

arguments

– (NSArray *)**arguments**

Returns the command line arguments as an array of NSStrings.

environment

– (NSDictionary *)**environment**

Returns a dictionary of variables for the environment from which the process was launched. The dictionary keys are the environment variable names.

globallyUniqueString

– (NSString *)**globallyUniqueString**

Returns a globally unique string to identify the process. This method uses the host name, process ID, and a time stamp to ensure that the string returned will be unique for the network. This method generates a new string each time it is invoked, so it also uses a counter to guarantee that strings created from the same process will be unique.

See also: – **processName:**

hostName

– (NSString *)**hostName**

Returns the name of the host system.

operatingSystem

– (unsigned int)**operatingSystem**

Returns one of the constants below to indicate the operating system on which the process is executing:

NSWindowsNTOperatingSystem
NSWindows95OperatingSystem
NSSolarisOperatingSystem
NSHPUXOperatingSystem
NSMACHOperatingSystem
NSSunOSOperatingSystem
NSOSF1OperatingSystem

processName

– (NSString *)**processName**

Returns the name of the process. This name is used to register Application defaults and is used in error messages. It does not uniquely identify the process.

See also: – **setProcessName:**

setProcessName:

– (void)**setProcessName:**(NSString *)*newName*

Sets the name of the process to *newName*.

Warning: User defaults and other aspects of the environment might depend on the process name, so be very careful if you change it. Setting the process name in this manner is **not** thread-safe.

See also: – **processName:**