NSProcessInfo Class Reference

Cocoa > Process Management

2007-03-26

Contents

NSProcessInfo Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Interacting with the Operating System
Declared in	NSProcessInfo.h
Related sample code	CocoaEcho CocoaHTTPServer Quartz Composer WWDC 2005 TextEdit Sproing TextEditPlus

Overview

The `NSProcessInfo` class provides methods to access information about the current process. Each process has a single, shared `NSProcessInfo` object, known as **process information agent**.

The process information agent can return such information as the arguments, environment variables, host name, or process name. The `processInfo` (page 7) class method returns the shared agent for the current process—that is, the process whose object sent the message. For example, the following line returns the `NSProcessInfo` object, which then provides the name of the current process:

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

The `NSProcessInfo` class also includes the `operatingSystem` (page 9) method, which returns an `enum` constant identifying the operating system on which the process is executing.

`NSProcessInfo` objects attempt to interpret environment variables and command-line arguments in the user's default C string encoding if they cannot be converted to Unicode as UTF-8 strings. If neither conversion works, these values are ignored by the `NSProcessInfo` object.

Tasks

Getting the Process Information Agent

`+ processInfo` (page 7)
: Returns the process information agent for the process.

Accessing Process Information

`- arguments` (page 8)
: Returns the command-line arguments for the process.

`- environment` (page 8)
: Returns the variable names and their values in the environment from which the process was launched.

`- processIdentifier` (page 10)
: Returns the identifier of the process.

`- globallyUniqueString` (page 8)
: Returns a global unique identifier for the process.

`- processName` (page 11)
: Returns the name of the process.

`- setProcessName:` (page 11)
: Sets the name of the process.

Getting Host Information

`- hostName` (page 9)
: Returns the name of the host computer.

`- operatingSystem` (page 9)
: Returns a constant to indicate the operating system on which the process is executing.

`- operatingSystemName` (page 9)
: Returns a string containing the name of the operating system on which the process is executing.

`- operatingSystemVersionString` (page 10)
: Returns a string containing the version of the operating system on which the process is executing.

Getting Computer Information

`- physicalMemory` (page 10)
: Provides the amount of physical memory on the computer.

`- processorCount` (page 11)
: Provides the number of processing cores available on the computer.

`- activeProcessorCount` (page 7)
: Provides the number of active processing cores available on the computer.

Class Methods

processInfo

Returns the process information agent for the process.

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

Return Value
Shared process information agent for the process.

Discussion
An `NSProcessInfo` (page 5) object is created the first time this method is invoked, and that same object is returned on each subsequent invocation.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
CocoaEcho

CocoaHTTPServer

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

URL CacheInfo

Declared In
`NSProcessInfo.h`

Instance Methods

activeProcessorCount

Provides the number of active processing cores available on the computer.

```
- (NSUInteger)activeProcessorCount
```

Return Value
Number of active processing cores.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- processorCount` (page 11)

Declared In
`NSProcessInfo.h`

arguments

Returns the command-line arguments for the process.

```
- (NSArray *)arguments
```

Return Value
Array of strings with the process's command-line arguments.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSProcessInfo.h`

environment

Returns the variable names and their values in the environment from which the process was launched.

```
- (NSDictionary *)environment
```

Return Value
Dictionary of environment-variable names (keys) and their values.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSProcessInfo.h`

globallyUniqueString

Returns a global unique identifier for the process.

```
- (NSString *)globallyUniqueString
```

Return Value
Global ID for the process. The ID includes the host name, process ID, and a time stamp, which ensures that the ID is unique for the network.

Discussion
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 are unique.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- processName` (page 11)

Related Sample Code
Quartz Composer WWDC 2005 TextEdit
TextEditPlus

Declared In
`NSProcessInfo.h`

hostName

Returns the name of the host computer.

```
- (NSString *)hostName
```

Return Value
Host name of the computer.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSProcessInfo.h`

operatingSystem

Returns a constant to indicate the operating system on which the process is executing.

```
- (unsigned int)operatingSystem
```

Return Value
Operating system identifier. See "Constants" (page 12) for a list of possible values. In Mac OS X, it's `NSMACHOperatingSystem`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSProcessInfo.h`

operatingSystemName

Returns a string containing the name of the operating system on which the process is executing.

```
- (NSString *)operatingSystemName
```

Return Value
Operating system name. In Mac OS X, it's `@"NSMACHOperatingSystem"`

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSProcessInfo.h`

operatingSystemVersionString

Returns a string containing the version of the operating system on which the process is executing.

```
- (NSString *)operatingSystemVersionString
```

Return Value
Operating system version. This string is human readable, localized, and is appropriate for displaying to the user. This string is *not* appropriate for parsing.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`NSProcessInfo.h`

physicalMemory

Provides the amount of physical memory on the computer.

```
- (unsigned long long)physicalMemory
```

Return Value
Amount of physical memory in bytes.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSProcessInfo.h`

processIdentifier

Returns the identifier of the process.

```
- (int)processIdentifier
```

Return Value
Process ID of the process.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- processName` (page 11)

Related Sample Code
Quartz Composer WWDC 2005 TextEdit
TextEditPlus

Declared In
`NSProcessInfo.h`

processName

Returns the name of the process.

```
- (NSString *)processName
```

Return Value
Name of the process.

Discussion
The process name is used to register application defaults and is used in error messages. It does not uniquely identify the process.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- processIdentifier` (page 10)
`- setProcessName:` (page 11)

Related Sample Code
URL CacheInfo

Declared In
`NSProcessInfo.h`

processorCount

Provides the number of processing cores available on the computer.

```
- (NSUInteger)processorCount
```

Return Value
Number of processing cores.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- activeProcessorCount` (page 7)

Declared In
`NSProcessInfo.h`

setProcessName:

Sets the name of the process.

```
- (void)setProcessName:(NSString *)name
```

Parameters

name

New name for the process.

Discussion

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.

Availability
Available in Mac OS X v10.0 and later.

See Also
`– processName` (page 11)

Declared In
`NSProcessInfo.h`

Constants

NSProcessInfo—Operating Systems

The following constants are provided by the `NSProcessInfo` class as return values for `operatingSystem` (page 9).

```
enum {
   NSWindowsNTOperatingSystem = 1,
   NSWindows95OperatingSystem,
   NSSolarisOperatingSystem,
   NSHPUXOperatingSystem,
   NSMACHOperatingSystem,
   NSSunOSOperatingSystem,
   NSOSF1OperatingSystem
};
```

Constants

`NSHPUXOperatingSystem`

Indicates the HP UX operating system.

Available in Mac OS X v10.0 and later.

Declared in `NSProcessInfo.h`.

`NSMACHOperatingSystem`

Indicates the Mac OS X operating system.

Available in Mac OS X v10.0 and later.

Declared in `NSProcessInfo.h`.

`NSOSF1OperatingSystem`

Indicates the OSF/1 operating system.

Available in Mac OS X v10.0 and later.

Declared in `NSProcessInfo.h`.

`NSSolarisOperatingSystem`
: Indicates the Solaris operating system.

  Available in Mac OS X v10.0 and later.

  Declared in `NSProcessInfo.h`.

`NSSunOSOperatingSystem`
: Indicates the Sun OS operating system.

  Available in Mac OS X v10.0 and later.

  Declared in `NSProcessInfo.h`.

`NSWindows95OperatingSystem`
: Indicates the Windows 95 operating system.

  Available in Mac OS X v10.0 and later.

  Declared in `NSProcessInfo.h`.

`NSWindowsNTOperatingSystem`
: Indicates the Windows NT operating system.

  Available in Mac OS X v10.0 and later.

  Declared in `NSProcessInfo.h`.

Declared In
`NSProcessInfo.h`

Document Revision History

This table describes the changes to *NSProcessInfo Class Reference*.

Date	Notes
2007-03-26	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index