Environment Class

Reference

Definition

Namespace: System

Assembly: System.Runtime.dll

Source: Environment.cs

Provides information about, and means to manipulate, the current environment and platform. This class cannot be inherited.

C#

```
public static class Environment
```

Inheritance Object → Environment

Examples

The following example displays a list of information about the current environment.

C#

```
// Sample for Environment class summary
using System;
using System.Collections;

class Sample
{
    public static void Main()
    {
        string str;
        string nl = Environment.NewLine;
        //
        Console.WriteLine();
        Console.WriteLine("-- Environment members --");

        //  Invoke this sample with an arbitrary set of command line
arguments.
        Console.WriteLine("CommandLine: {0}",
Environment.CommandLine);

        string[] arguments = Environment.GetCommandLineArgs();
        Console.WriteLine("GetCommandLineArgs: {0}", String.Join(",
```

```
", arguments));

        //  <-- Keep this information secure! -->
        Console.WriteLine("CurrentDirectory: {0}",
Environment.CurrentDirectory);

        Console.WriteLine("ExitCode: {0}", Environment.ExitCode);

        Console.WriteLine("HasShutdownStarted: {0}",
Environment.HasShutdownStarted);

        //  <-- Keep this information secure! -->
        Console.WriteLine("MachineName: {0}",
Environment.MachineName);

        Console.WriteLine("NewLine: {0}  first line{0}  second
line{0}  third line",
                          Environment.NewLine);

        Console.WriteLine("OSVersion: {0}",
Environment.OSVersion.ToString());

        Console.WriteLine("StackTrace: '{0}'",
Environment.StackTrace);

        //  <-- Keep this information secure! -->
        Console.WriteLine("SystemDirectory: {0}",
Environment.SystemDirectory);

        Console.WriteLine("TickCount: {0}", Environment.TickCount);

        //  <-- Keep this information secure! -->
        Console.WriteLine("UserDomainName: {0}",
Environment.UserDomainName);

        Console.WriteLine("UserInteractive: {0}",
Environment.UserInteractive);

        //  <-- Keep this information secure! -->
        Console.WriteLine("UserName: {0}", Environment.UserName);

        Console.WriteLine("Version: {0}",
Environment.Version.ToString());

        Console.WriteLine("WorkingSet: {0}", Environment.WorkingSet);

        //  No example for Exit(exitCode) because doing so would ter-
minate this example.

        //  <-- Keep this information secure! -->
        string query = "My system drive is %SystemDrive% and my sys-
tem root is %SystemRoot%";
        str = Environment.ExpandEnvironmentVariables(query);
        Console.WriteLine("ExpandEnvironmentVariables: {0}  {1}", nl,
str);
```

```
        Console.WriteLine("GetEnvironmentVariable: {0}  My temporary
directory is {1}.", nl,

Environment.GetEnvironmentVariable("TEMP"));

        Console.WriteLine("GetEnvironmentVariables: ");
        IDictionary environmentVariables =
Environment.GetEnvironmentVariables();
        foreach (DictionaryEntry de in environmentVariables)
        {
            Console.WriteLine("  {0} = {1}", de.Key, de.Value);
        }

        Console.WriteLine("GetFolderPath: {0}",

Environment.GetFolderPath(Environment.SpecialFolder.System));

        string[] drives = Environment.GetLogicalDrives();
        Console.WriteLine("GetLogicalDrives: {0}", String.Join(", ",
drives));
    }
}
/*
This example produces results similar to the following:
(Any result that is lengthy or reveals information that should remain
secure has been omitted and marked "!---OMITTED---!".)

C:\>env0 ARBITRARY TEXT

-- Environment members --
CommandLine: env0 ARBITRARY TEXT
GetCommandLineArgs: env0, ARBITRARY, TEXT
CurrentDirectory: C:\Documents and Settings\!---OMITTED---!
ExitCode: 0
HasShutdownStarted: False
MachineName: !---OMITTED---!
NewLine:
  first line
  second line
  third line
OSVersion: Microsoft Windows NT 5.1.2600.0
StackTrace: '   at System.Environment.GetStackTrace(Exception e)
   at System.Environment.GetStackTrace(Exception e)
   at System.Environment.get_StackTrace()
   at Sample.Main()'
SystemDirectory: C:\WINNT\System32
TickCount: 17995355
UserDomainName: !---OMITTED---!
UserInteractive: True
UserName: !---OMITTED---!
Version: !---OMITTED---!
WorkingSet: 5038080
ExpandEnvironmentVariables:
  My system drive is C: and my system root is C:\WINNT
```

```
GetEnvironmentVariable:
  My temporary directory is C:\DOCUME~1\!---OMITTED--
-!\LOCALS~1\Temp.
GetEnvironmentVariables:
  !---OMITTED---!
GetFolderPath: C:\WINNT\System32
GetLogicalDrives: A:\, C:\, D:\

*/
```

Remarks

Use the Environment class to retrieve information such as command-line arguments, the exit code, environment variable settings, contents of the call stack, time since last system boot, and the version of the common language runtime.

Properties

Expand table

CommandLine	Gets the command line for this process.
CurrentDirectory	Gets or sets the fully qualified path of the current working directory.
CurrentManagedThreadId	Gets a unique identifier for the current managed thread.
ExitCode	Gets or sets the exit code of the process.
HasShutdownStarted	Gets a value that indicates whether the current application domain is being unloaded or the common language runtime (CLR) is shutting down.
Is64BitOperatingSystem	Gets a value that indicates whether the current operating system is a 64-bit operating system.
Is64BitProcess	Gets a value that indicates whether the current process is a 64-bit process.
IsPrivilegedProcess	Gets a value that indicates whether the current process is authorized to perform security-relevant functions.
MachineName	Gets the NetBIOS name of this local computer.
NewLine	Gets the newline string defined for this environment.
OSVersion	Gets the current platform identifier and version number.
ProcessId	Gets the unique identifier for the current process.

ProcessorCount	Gets the number of processors available to the current process.
ProcessPath	Returns the path of the executable that started the currently executing process. Returns `null` when the path is not available.
StackTrace	Gets current stack trace information.
SystemDirectory	Gets the fully qualified path of the system directory.
SystemPageSize	Gets the number of bytes in the operating system's memory page.
TickCount	Gets the number of milliseconds elapsed since the system started.
TickCount64	Gets the number of milliseconds elapsed since the system started.
UserDomainName	Gets the network domain name associated with the current user.
UserInteractive	Gets a value indicating whether the current process is running in user interactive mode.
UserName	Gets the user name of the person who is associated with the current thread.
Version	Gets a version consisting of the major, minor, build, and revision numbers of the common language runtime.
WorkingSet	Gets the amount of physical memory mapped to the process context.

Methods

Expand table

Exit(Int32)	Terminates this process and returns an exit code to the operating system.
ExpandEnvironment Variables(String)	Replaces the name of each environment variable embedded in the specified string with the string equivalent of the value of the variable, then returns the resulting string.
FailFast(String)	Immediately terminates the process before reporting an error message. For Windows, the error message is written to the Windows Application event log, and the message is included in error reporting to Microsoft. For Unix-like systems, the message, alongside the stack trace, is written to the standard error stream.
FailFast(String, Exception)	Immediately terminates the process before reporting an error message. For Windows, the error message is written to the Windows Application event log, and the message and exception information is included in error reporting to

	Microsoft. For Unix-like systems, the message alongside the stack trace is written to the standard error stream.
GetCommandLineArgs()	Returns a string array containing the command-line arguments for the current process.
GetEnvironmentVariable(String)	Retrieves the value of an environment variable from the current process.
GetEnvironmentVariable(String, EnvironmentVariableTarget)	Retrieves the value of an environment variable from the current process or from the Windows operating system registry key for the current user or local machine.
GetEnvironmentVariables()	Retrieves all environment variable names and their values from the current process.
GetEnvironment Variables(EnvironmentVariable Target)	Retrieves all environment variable names and their values from the current process, or from the Windows operating system registry key for the current user or local machine.
GetFolder Path(Environment+SpecialFolder)	Gets the path to the specified system special folder.
GetFolder Path(Environment+SpecialFolder, Environment+SpecialFolderOption)	Gets the path to the specified system special folder using a specified option for accessing special folders.
GetLogicalDrives()	Returns an array of string containing the names of the logical drives on the current computer.
SetEnvironmentVariable(String, String)	Creates, modifies, or deletes an environment variable stored in the current process.
SetEnvironmentVariable(String, String, EnvironmentVariableTarget)	Creates, modifies, or deletes an environment variable stored in the current process or in the Windows operating system registry key reserved for the current user or local machine.

Applies to

Product	Versions
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8, 9
.NET Framework	1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8, 4.8.1
.NET Standard	1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 2.0, 2.1
UWP	10.0

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

.NET feedback

.NET is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback