

Forum:[Hack Forums](#)

Thread: [Get All Environment Variables - 2 Python Methods / Scripts - FREE Source Code](#)

Hackers - May 1, 2024, 7:09 am

Get All Environment Variables 2 Python Methods / Scripts - FREE Source Code

In the first example below, we use the 'os' package to get environment variables. It's easy to see what's going on because of how the package already handles all of the hard work for us.

Code import os

```
# Get all environment variables
env_vars = os.environ
```

```
# Open file to write
with open('output.txt', 'w') as f:
    # Iterate over each environment variable
    for key, value in env_vars.items():
        # Write to file
        f.write(f'{key}: {value}\n')
```

Example output which gives all of the key=value dumps:

Spoiler ([Click to View](#))

```
Code ALLUSERSPROFILE: C:\ProgramData
APPDATA: C:\Users\nodes\AppData\Roaming
CHOCOLATEYINSTALL: C:\ProgramData\chocolatey
CHOCOLATEYLASTPATHUPDATE: 133516425516584110
COMMONPROGRAMFILES: C:\Program Files\Common Files
COMMONPROGRAMFILES(X86): C:\Program Files (x86)\Common Files
COMMONPROGRAMW6432: C:\Program Files\Common Files
COMPUTERNAME: DESKTOP-nodes
COMSPEC: C:\WINDOWS\system32\cmd.exe
CPLUS_INCLUDE_PATH:
C:\ProgramData\chocolatey\lib\curl\include;%CPLUS_INCLUDE_PATH%
DRIVERDATA: C:\Windows\System32\Drivers\DriverData
EFC_11384: 1
FPS_BROWSER_APP_PROFILE_STRING: Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING: Default
HOME: C:\Users\nodes
HOMEDRIVE: C:
HOMEPATH: \Users\nodes
LIBRARY_PATH: C:\ProgramData\chocolatey\lib\curl\lib;%LIBRARY_PATH%
LOCALAPPDATA: C:\Users\nodes\AppData\Local
LOGONSERVER: \\DESKTOP-nodes
NUMBER_OF_PROCESSORS: 8
NVM_HOME: C:\Users\nodes\AppData\Roaming\nvm
NVM_SYMLINK: C:\Program Files\nodejs
ONEDRIVE: C:\Users\nodes\OneDrive
```

```

OPENSSL_CONF: C:\Program Files\OpenSSL-Win64\bin\openssl.cfg
OS: Windows_NT
PATH: C:\Program Files (x86)\Common
Files\Oracle\Java\javapath;C:\Python312\Scripts;C:\Python312;C:\Python311\Scripts;C:\Python31
1;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System3
2\WindowsPowerShell\v1.0;C:\WINDOWS\System32\OpenSSH;C:\Program
Files\PuTTY;C:\ProgramData\chocolatey\bin;C:\Program
Files\Docker\Docker\resources\bin;C:\ProgramData\DockerDesktop\version-bin;C:\Program
Files\Git\cmd;C:\Users\nodes\AppData\Local\Android\Sdk\platform-tools;C:\Users\nodes\AppData\R
oaming\nvm;C:\Program Files\nodejs;C:\Qt\6.6.0\mingw_64\bin;C:\Program Files
(x86)\nodejs;C:\Program Files (x86)\dotnet;C:\Python312\Scripts;C:\Program Files (x86)\Windows
Kits\10\Windows Performance Toolkit;C:\Program
Files\dotnet;C:\Ruby32-x64\bin;C:\Python310\Scripts;C:\Python310;C:\Users\nodes\AppData\Loca
l\Programs\Python\Python310\Scripts;C:\Users\nodes\AppData\Local\Programs\Python\Python310\
;C:\Users\nodes\AppData\Local\Microsoft\WindowsApps;C:\Users\nodes\AppData\Local\GitHubDes
ktop\bin;C:\Users\nodes\AppData\Local\Programs\Microsoft VS Code\bin;C:\Program Files
(x86)\BrowserStackLocal;C:\Users\nodes\AppData\Roaming\nvm;C:\Program
Files\nodejs;C:\ProgramData\mingw64\mingw64\bin;C:\Program
Files\CMake\bin;C:\Qt\5.15.2\mingw81_64\bin;C:\Users\nodes\AppData\Roaming\npm;C:\Python31
2;C:\Program
Files\OpenSSL-Win64\bin;C:\Users\nodes\AppData\Local\bin\NASM;C:\Users\nodes\dotnet\tools;C
:\Program Files\Google\Chrome\Application;
PATHEXT: .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW;.RB;.RBW
PROCESSOR_ARCHITECTURE: AMD64
PROCESSOR_IDENTIFIER: Intel64 Family 6 Model 142 Stepping 12, GenuineIntel
PROCESSOR_LEVEL: 6
PROCESSOR_REVISION: 8e0c
PROGRAMDATA: C:\ProgramData
PROGRAMFILES: C:\Program Files
PROGRAMFILES(X86): C:\Program Files (x86)
PROGRAMW6432: C:\Program Files
PSMODULEPATH: C:\Program
Files\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
PUBLIC: C:\Users\Public
SESSIONNAME: Console
SYSTEMDRIVE: C:
SYSTEMROOT: C:\WINDOWS
TEMP: C:\Users\nodes\AppData\Local\Temp
TMP: C:\Users\nodes\AppData\Local\Temp
USERDOMAIN: DESKTOP-nodes
USERDOMAIN_ROAMINGPROFILE: DESKTOP-nodes
USERNAME: nodes
USERPROFILE: C:\Users\nodes
VBOX_MSI_INSTALL_PATH: C:\Program Files\Oracle\VirtualBox\
WINDIR: C:\WINDOWS

```

In the next example, we go to a lower level and use the ctypes package to get wintypes & use kernel32 to fetch environment variables (GetEnvironmentVariableW). Just add or remove properties from the list if you need different data.

```

Code import ctypes
from ctypes.wintypes import LPWSTR, DWORD

```

```

# Load the kernel32.dll

```

```
kernel32 = ctypes.WinDLL('kernel32', use_last_error=True)
```

```
# Define the GetEnvironmentVariable function
get_env_var = kernel32.GetEnvironmentVariableW
get_env_var.argtypes = [LPWSTR, LPWSTR, DWORD]
get_env_var.restype = DWORD
```

```
def get_environment_variable(name):
    buffer = ctypes.create_unicode_buffer(32767) # Large buffer for the environment variable
    ret = get_env_var(name, buffer, 32767)
    if ret == 0:
        error = ctypes.get_last_error()
        if error:
            raise ctypes.WinError(error)
    return buffer.value
```

```
# List of environment variables to fetch
env_vars = [
    'ALLUSERSPROFILE', 'APPDATA', 'CHOCOLATEYINSTALL',
    'CHOCOLATEYLASTPATHUPDATE',
    'COMMONPROGRAMFILES', 'COMMONPROGRAMFILES(X86)',
    'COMMONPROGRAMW6432', 'COMPUTERNAME',
    'COMSPEC', 'CPLUS_INCLUDE_PATH', 'DRIVERDATA', 'EFC_11384',
    'FPS_BROWSER_APP_PROFILE_STRING',
    'FPS_BROWSER_USER_PROFILE_STRING', 'HOME', 'HOMEDRIVE', 'HOMEPATH',
    'LIBRARY_PATH',
    'LOCALAPPDATA', 'LOGONSERVER', 'NUMBER_OF_PROCESSORS', 'NVM_HOME',
    'NVM_SYMLINK',
    'ONEDRIVE', 'OPENSSL_CONF', 'OS', 'PATH', 'PATHEXT', 'PROCESSOR_ARCHITECTURE',
    'PROCESSOR_IDENTIFIER', 'PROCESSOR_LEVEL', 'PROCESSOR_REVISION',
    'PROGRAMDATA',
    'PROGRAMFILES', 'PROGRAMFILES(X86)', 'PROGRAMW6432', 'PSMODULEPATH', 'PUBLIC',

    'SESSIONNAME', 'SYSTEMDRIVE', 'SYSTEMROOT', 'TEMP', 'TMP', 'USERDOMAIN',
    'USERDOMAIN_ROAMINGPROFILE', 'USERNAME', 'USERPROFILE',
    'VBOX_MSI_INSTALL_PATH', 'WINDIR'
]
```

```
# Fetch the values
env_values = {var: get_environment_variable(var) for var in env_vars}
```

```
# Write the fetched environment variables to 'output.txt'
with open('output.txt', 'w') as file:
```

```
    for key, value in env_values.items():
        file.write(f'{key}: {value}\n')
```

Here is our output.txt file from the ctypes Python script:

Spoiler ([Click to View](#))

```
Code ALLUSERSPROFILE: C:\ProgramData
APPDATA: C:\Users\nodes\AppData\Roaming
CHOCOLATEYINSTALL: C:\ProgramData\chocolatey
CHOCOLATEYLASTPATHUPDATE: 133516425516584110
COMMONPROGRAMFILES: C:\Program Files\Common Files
COMMONPROGRAMFILES(X86): C:\Program Files (x86)\Common Files
```

COMMONPROGRAMW6432: C:\Program Files\Common Files
COMPUTERNAME: DESKTOP-nodes
COMSPEC: C:\WINDOWS\system32\cmd.exe
CPLUS_INCLUDE_PATH:
C:\ProgramData\chocolatey\lib\curl\include;%CPLUS_INCLUDE_PATH%
DRIVERDATA: C:\Windows\System32\Drivers\DriverData
EFC_11384: 1
FPS_BROWSER_APP_PROFILE_STRING: Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING: Default
HOME: C:\Users\nodes
HOMEDRIVE: C:
HOMEPATH: \Users\nodes
LIBRARY_PATH: C:\ProgramData\chocolatey\lib\curl\lib;%LIBRARY_PATH%
LOCALAPPDATA: C:\Users\nodes\AppData\Local
LOGONSERVER: \\DESKTOP-nodes
NUMBER_OF_PROCESSORS: 8
NVM_HOME: C:\Users\nodes\AppData\Roaming\nvm
NVM_SYMLINK: C:\Program Files\nodejs
ONEDRIVE: C:\Users\nodes\OneDrive
OPENSSL_CONF: C:\Program Files\OpenSSL-Win64\bin\openssl.cfg
OS: Windows_NT
PATH: C:\Program Files (x86)\Common
Files\Oracle\Java\javapath;C:\Python312\Scripts\;C:\Python312\;C:\Python311\Scripts\;C:\Python311\;
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;
C:\WINDOWS\System32\OpenSSH\;C:\Program Files\PuTTY\;C:\ProgramData\chocolatey\bin;C:\Program
Files\Docker\Docker\resources\bin;C:\ProgramData\DockerDesktop\version-bin;C:\Program
Files\Git\cmd;C:\Users\nodes\AppData\Local\Android\Sdk\platform-tools;C:\Users\nodes\AppData\Roaming\nvm;
C:\Program Files\nodejs;C:\Qt\6.6.0\mingw_64\bin;C:\Program Files (x86)\nodejs\;C:\Program Files (x86)\dotnet\;
C:\Python312\Scripts;C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\;C:\Program
Files\dotnet\;C:\Ruby32-x64\bin;C:\Python310\Scripts\;C:\Python310\;C:\Users\nodes\AppData\Local\Programs\Python\Python310\Scripts\;
C:\Users\nodes\AppData\Local\Programs\Python\Python310\;C:\Users\nodes\AppData\Local\Microsoft\WindowsApps;
C:\Users\nodes\AppData\Local\GitHubDesktop\bin;C:\Users\nodes\AppData\Local\Programs\Microsoft VS Code\bin;C:\Program Files (x86)\BrowserStackLocal\;
C:\Users\nodes\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\ProgramData\mingw64\mingw64\bin;C:\Program
Files\CMake\bin;C:\Qt\5.15.2\mingw81_64\bin;C:\Users\nodes\AppData\Roaming\npm;C:\Python312\;C:\Program
Files\OpenSSL-Win64\bin;C:\Users\nodes\AppData\Local\bin\NASM;C:\Users\nodes\dotnet\tools;C:\Program Files\Google\Chrome\Application;
PATHEXT: .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW;.RB;.RBW
PROCESSOR_ARCHITECTURE: AMD64
PROCESSOR_IDENTIFIER: Intel64 Family 6 Model 142 Stepping 12, GenuineIntel
PROCESSOR_LEVEL: 6
PROCESSOR_REVISION: 8e0c
PROGRAMDATA: C:\ProgramData
PROGRAMFILES: C:\Program Files
PROGRAMFILES(X86): C:\Program Files (x86)
PROGRAMW6432: C:\Program Files
PSMODULEPATH: C:\Program
Files\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
PUBLIC: C:\Users\Public

SESSIONNAME: Console
SYSTEMDRIVE: C:
SYSTEMROOT: C:\WINDOWS
TEMP: C:\Users\nodes\AppData\Local\Temp
TMP: C:\Users\nodes\AppData\Local\Temp
USERDOMAIN: DESKTOP-nodes
USERDOMAIN_ROAMINGPROFILE: DESKTOP-nodes
USERNAME: nodes
USERPROFILE: C:\Users\nodes
VBOX_MSI_INSTALL_PATH: C:\Program Files\Oracle\VirtualBox\
WINDIR: C:\WINDOWS

flyingduck - May 1, 2024, 7:13 am

looking good! thx for putting up another quality share bro! <3

Hackers - May 1, 2024, 8:36 am

(May 1, 2024 07:13 AM)flyingduck Wrote: [\[CITE\]](#) looking good! thx for putting up another quality share bro! <3

thanks duck! Hope this helps someone, it can be useful for a lot of different reasons.

If anyone wants help understanding part of the code just let me know!

flyingduck - May 1, 2024, 9:00 am

(May 1, 2024 08:36 AM)nodes Wrote: [\[CITE\]](#) thanks duck! Hope this helps someone, it can be useful for a lot of different reasons.

If anyone wants help understanding part of the code just let me know!
done and done 4 u :P

This Python script performs a couple of major functions related to working with environment variables on a Windows system, including accessing and writing them to a file. Here's a breakdown of its components and their purposes:

Importing Modules:

os: This module provides a way to interact with the operating system. In this script, it's used to access environment variables.

ctypes and ctypes.wintypes: These are used to interact with the Windows API, allowing for deeper system-level interactions.

Extracting and Writing Environment Variables to a File:

The script initially uses `os.environ` to get all environment variables, storing them in `env_vars`. It then opens a file named `output.txt` in write mode.

It iterates over the `env_vars` dictionary and writes each key-value pair (representing environment variable names and values) to the file in the format `KEY: VALUE`.

Loading kernel32.dll Using ctypes:

`kernel32 = ctypes.WinDLL('kernel32', use_last_error=True):` This line loads the `kernel32.dll` library (a core Windows library) into the script, enabling direct API calls to system-level functions.

The script sets up a function `GetEnvironmentVariableW` from this DLL to fetch environment variable values, tailored to handle wide character strings (Unicode).

Defining a Python Function to Fetch Environment Variables:

`get_environment_variable:` This function is defined to wrap the `GetEnvironmentVariableW` API

call. It uses a large buffer to ensure it can handle the value of any environment variable.

It checks if the API call fails (returns 0), and if so, it raises an error with the last error reported by the system.

Fetching Specific Environment Variables:

A list of specific environment variable names is defined.

The script then iterates over this list, fetching each environment variable's value using the previously defined `get_environment_variable` function.

Writing Specific Environment Variables to a File:

The script opens `output.txt` again in write mode and writes the values of the specific environment variables fetched via the Windows API.

This script is particularly useful for debugging or managing environment variables in a Windows environment, and it demonstrates advanced usage of Python for interacting with Windows system functions via APIs.

LLynn101 - May 1, 2024, 9:33 am

Can you please recommend me to an AI app developer. I have a project I would like to get off the ground

Hackers - May 2, 2024, 4:32 am

(May 1, 2024 09:00 AM)flyingduck Wrote: [\[CITE\]](#) done and done 4 u :P

(May 1, 2024 09:33 AM)LLynn101 Wrote: [\[CITE\]](#) Can you please recommend me to an AI app developer. I have a project I would like to get off the ground

Nice... What was your favorite part of this thread?

G.P.D. - May 5, 2024, 8:15 am

One-liner in Python for SSTI :

```
Code open('leak.env', 'w').write(__import__('json').dumps(dict(__import__('os').environ), indent=4))
```

K24Barton - June 11, 2024, 7:10 pm

I'll come back when I know more Python.

Hackers - June 11, 2024, 7:17 pm

(Jun 11, 2024 07:10 PM)K24Barton Wrote: [\[CITE\]](#) I'll come back when I know more Python.

nice, hope to see you soon!

Best of luck, let me know if you have any questions

Stanley - March 16, 2026, 8:31 pm

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley