Python winreg.OpenKey() Examples

The following are *50* code examples for showing how to use *winreg.OpenKey()*. They are extracted from open source Python projects. You can vote up the examples you like or vote down the exmaples you don't like. You can also save this page to your account.

**+** Save to library

**Example 1**

|  |  |  |
| --- | --- | --- |
| Project: *get-youtube-subtitle-url-node*   Author: *joegesualdo*   File: *buildserver.py*    [(license)](https://github.com/joegesualdo/get-youtube-subtitle-url-node)[*View Source Project*](https://github.com/joegesualdo/get-youtube-subtitle-url-node/tree/master/youtube-dl/devscripts/buildserver.py) | 9 votes | vote downvote up |

def \_\_init\_\_(self, \*\*kwargs):

python\_version = kwargs.pop('python', '3.4')

python\_path = None

for node in ('Wow6432Node\\', ''):

try:

key = compat\_winreg.OpenKey(

compat\_winreg.HKEY\_LOCAL\_MACHINE,

r'SOFTWARE\%sPython\PythonCore\%s\InstallPath' % (node, python\_version))

try:

python\_path, \_ = compat\_winreg.QueryValueEx(key, '')

finally:

compat\_winreg.CloseKey(key)

break

except Exception:

pass

if not python\_path:

raise BuildError('No such Python version: %s' % python\_version)

self.pythonPath = python\_path

super(PythonBuilder, self).\_\_init\_\_(\*\*kwargs)

**Example 2**

|  |  |  |
| --- | --- | --- |
| Project: *pip-update-requirements*   Author: *alanhamlett*   File: *appdirs.py*    [(BSD 2-Clause "Simplified" License)](https://github.com/alanhamlett/pip-update-requirements)[*View Source Project*](https://github.com/alanhamlett/pip-update-requirements/tree/master/pur/packages/pip/_vendor/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 3**

|  |  |  |
| --- | --- | --- |
| Project: *paste2box*   Author: *rokups*   File: *settings.py*    [(GNU General Public License v3.0)](https://github.com/rokups/paste2box)[*View Source Project*](https://github.com/rokups/paste2box/tree/master/gui/settings.py) | 6 votes | vote downvote up |

def is\_autostart\_enabled(self):

if os.name == 'nt':

with winreg.OpenKey(winreg.HKEY\_CURRENT\_USER, self.\_windows\_run\_reg\_key, 0, winreg.KEY\_ALL\_ACCESS) as key:

try:

reg\_value, reg\_type = winreg.QueryValueEx(key, const.APP\_NAME)

if reg\_type == winreg.REG\_SZ and reg\_value == self.\_get\_executable\_path():

return True

else:

try:

winreg.DeleteValue(key, self.WIN\_REG\_AUTORUN\_KEY)

except OSError:

pass # key does not exist

except:

return False

else:

return os.path.exists(self.\_linux\_autostart\_file)

**Example 4**

|  |  |  |
| --- | --- | --- |
| Project: *muesr*   Author: *bonfus*   File: *appdirs.py*    [(license)](https://github.com/bonfus/muesr)[*View Source Project*](https://github.com/bonfus/muesr/tree/master/muesr/core/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 5**

|  |  |  |
| --- | --- | --- |
| Project: *ivaochdoc*   Author: *ivaoch*   File: *appdirs.py*    [(license)](https://github.com/ivaoch/ivaochdoc)[*View Source Project*](https://github.com/ivaoch/ivaochdoc/tree/master/mkdocs_venv/Lib/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 6**

|  |  |  |
| --- | --- | --- |
| Project: *optimalvibes*   Author: *littlemika*   File: *buildserver.py*    [(license)](https://github.com/littlemika/optimalvibes)[*View Source Project*](https://github.com/littlemika/optimalvibes/tree/master/server/packages/youtube-dl/devscripts/buildserver.py) | 6 votes | vote downvote up |

def \_\_init\_\_(self, \*\*kwargs):

python\_version = kwargs.pop('python', '3.4')

python\_path = None

for node in ('Wow6432Node\\', ''):

try:

key = compat\_winreg.OpenKey(

compat\_winreg.HKEY\_LOCAL\_MACHINE,

r'SOFTWARE\%sPython\PythonCore\%s\InstallPath' % (node, python\_version))

try:

python\_path, \_ = compat\_winreg.QueryValueEx(key, '')

finally:

compat\_winreg.CloseKey(key)

break

except Exception:

pass

if not python\_path:

raise BuildError('No such Python version: %s' % python\_version)

self.pythonPath = python\_path

super(PythonBuilder, self).\_\_init\_\_(\*\*kwargs)

**Example 7**

|  |  |  |
| --- | --- | --- |
| Project: *RealtimePythonChat*   Author: *quangtqag*   File: *appdirs.py*    [(license)](https://github.com/quangtqag/RealtimePythonChat)[*View Source Project*](https://github.com/quangtqag/RealtimePythonChat/tree/master/venv/lib/python2.7/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 8**

|  |  |  |
| --- | --- | --- |
| Project: *leetcode*   Author: *thomasyimgit*   File: *test\_path.py*    [(license)](https://github.com/thomasyimgit/leetcode)[*View Source Project*](https://github.com/thomasyimgit/leetcode/tree/master/env/lib/python3.6/site-packages/IPython/utils/tests/test_path.py) | 6 votes | vote downvote up |

def test\_get\_home\_dir\_8():

"""Using registry hack for 'My Documents', os=='nt'

HOMESHARE, HOMEDRIVE, HOMEPATH, USERPROFILE and others are missing.

"""

os.name = 'nt'

# Remove from stub environment all keys that may be set

for key in ['HOME', 'HOMESHARE', 'HOMEDRIVE', 'HOMEPATH', 'USERPROFILE']:

env.pop(key, None)

class key:

def Close(self):

pass

with patch.object(wreg, 'OpenKey', return\_value=key()), \

patch.object(wreg, 'QueryValueEx', return\_value=[abspath(HOME\_TEST\_DIR)]):

home\_dir = path.get\_home\_dir()

nt.assert\_equal(home\_dir, abspath(HOME\_TEST\_DIR))

**Example 9**

|  |  |  |
| --- | --- | --- |
| Project: *exoduscli*   Author: *cthlo*   File: *appdirs.py*    [(license)](https://github.com/cthlo/exoduscli)[*View Source Project*](https://github.com/cthlo/exoduscli/tree/master/exoduscli/lib/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 10**

|  |  |  |
| --- | --- | --- |
| Project: *gn-build*   Author: *timniederhausen*   File: *toolchain.py*    [(license)](https://github.com/timniederhausen/gn-build)[*View Source Project*](https://github.com/timniederhausen/gn-build/tree/master/toolchain/win/toolchain.py) | 6 votes | vote downvote up |

def \_RegistryGetValueUsingWinReg(key, value):

"""Use the \_winreg module to obtain the value of a registry key.

Args:

key: The registry key.

value: The particular registry value to read.

Return:

contents of the registry key's value, or None on failure. Throws

ImportError if \_winreg is unavailable.

"""

try:

import \_winreg

except ImportError:

import winreg as \_winreg

try:

root, subkey = key.split('\\', 1)

assert root == 'HKLM' # Only need HKLM for now.

with \_winreg.OpenKey(\_winreg.HKEY\_LOCAL\_MACHINE, subkey) as hkey:

return \_winreg.QueryValueEx(hkey, value)[0]

except WindowsError:

return None

**Example 11**

|  |  |  |
| --- | --- | --- |
| Project: *Tencent\_Cartoon\_Download*   Author: *Fretice*   File: *appdirs.py*    [(license)](https://github.com/Fretice/Tencent_Cartoon_Download)[*View Source Project*](https://github.com/Fretice/Tencent_Cartoon_Download/tree/master/venv/lib/python3.5/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 12**

|  |  |  |
| --- | --- | --- |
| Project: *coolrelation*   Author: *mrtial*   File: *appdirs.py*    [(license)](https://github.com/mrtial/coolrelation)[*View Source Project*](https://github.com/mrtial/coolrelation/tree/master/coolrelation/lib/python3.5/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 13**

|  |  |  |
| --- | --- | --- |
| Project: *cottoncandy*   Author: *gallantlab*   File: *appdirs.py*    [(license)](https://github.com/gallantlab/cottoncandy)[*View Source Project*](https://github.com/gallantlab/cottoncandy/tree/master/cottoncandy/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

import winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders")

dir, type = winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 14**

|  |  |  |
| --- | --- | --- |
| Project: *python-group-proj*   Author: *Sharcee*   File: *appdirs.py*    [(license)](https://github.com/Sharcee/python-group-proj)[*View Source Project*](https://github.com/Sharcee/python-group-proj/tree/master/venv/lib/python2.7/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 15**

|  |  |  |
| --- | --- | --- |
| Project: *PornGuys*   Author: *followloda*   File: *appdirs.py*    [(license)](https://github.com/followloda/PornGuys)[*View Source Project*](https://github.com/followloda/PornGuys/tree/master/FlaskServer/venv/Lib/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 16**

|  |  |  |
| --- | --- | --- |
| Project: *pyjam*   Author: *10se1ucgo*   File: *common.py*    [(license)](https://github.com/10se1ucgo/pyjam)[*View Source Project*](https://github.com/10se1ucgo/pyjam/tree/master/jam/common.py) | 6 votes | vote downvote up |

def get\_steam\_path():

"""Get the path for Steam from the Steam process. If that fails, it uses the registry on Windows.

Returns:

str: The path to Steam. If the path could not be found, the current directory is returned instead (os.curdir)

"""

if psutil:

for pid in psutil.process\_iter():

try:

if pid.name().lower() == 'steam.exe' or pid.name().lower() == 'steam':

return os.path.dirname(pid.exe())

except psutil.Error:

logger.exception("Could not get Steam path from its process.")

if winreg:

try:

reg\_key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER, r'Software\Valve\Steam')

return os.path.normpath(winreg.QueryValueEx(reg\_key, r'SteamPath')[0])

except WindowsError:

logger.exception("Could not query registry for Steam path")

return os.curdir

**Example 17**

|  |  |  |
| --- | --- | --- |
| Project: *Repobot*   Author: *Desgard*   File: *appdirs.py*    [(license)](https://github.com/Desgard/Repobot)[*View Source Project*](https://github.com/Desgard/Repobot/tree/master/repobot-lenv/lib/python3.6/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 18**

|  |  |  |
| --- | --- | --- |
| Project: *Repobot*   Author: *Desgard*   File: *test\_path.py*    [(license)](https://github.com/Desgard/Repobot)[*View Source Project*](https://github.com/Desgard/Repobot/tree/master/repobot-lenv/lib/python3.6/site-packages/IPython/utils/tests/test_path.py) | 6 votes | vote downvote up |

def test\_get\_home\_dir\_8():

"""Using registry hack for 'My Documents', os=='nt'

HOMESHARE, HOMEDRIVE, HOMEPATH, USERPROFILE and others are missing.

"""

os.name = 'nt'

# Remove from stub environment all keys that may be set

for key in ['HOME', 'HOMESHARE', 'HOMEDRIVE', 'HOMEPATH', 'USERPROFILE']:

env.pop(key, None)

class key:

def Close(self):

pass

with patch.object(wreg, 'OpenKey', return\_value=key()), \

patch.object(wreg, 'QueryValueEx', return\_value=[abspath(HOME\_TEST\_DIR)]):

home\_dir = path.get\_home\_dir()

nt.assert\_equal(home\_dir, abspath(HOME\_TEST\_DIR))

**Example 19**

|  |  |  |
| --- | --- | --- |
| Project: *CloudPrint*   Author: *William-An*   File: *appdirs.py*    [(license)](https://github.com/William-An/CloudPrint)[*View Source Project*](https://github.com/William-An/CloudPrint/tree/master/cloudprint/Lib/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 20**

|  |  |  |
| --- | --- | --- |
| Project: *pipenv*   Author: *pypa*   File: *appdirs.py*    [(license)](https://github.com/pypa/pipenv)[*View Source Project*](https://github.com/pypa/pipenv/tree/master/pipenv/vendor/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 21**

|  |  |  |
| --- | --- | --- |
| Project: *QualquerMerdaAPI*   Author: *tiagovizoto*   File: *appdirs.py*    [(license)](https://github.com/tiagovizoto/QualquerMerdaAPI)[*View Source Project*](https://github.com/tiagovizoto/QualquerMerdaAPI/tree/master/kap/lib/python3.4/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 22**

|  |  |  |
| --- | --- | --- |
| Project: *gardenbot*   Author: *GoestaO*   File: *appdirs.py*    [(license)](https://github.com/GoestaO/gardenbot)[*View Source Project*](https://github.com/GoestaO/gardenbot/tree/master/virtualenv/lib/python3.5/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 23**

|  |  |  |
| --- | --- | --- |
| Project: *hate-to-hugs*   Author: *sdoran35*   File: *appdirs.py*    [(license)](https://github.com/sdoran35/hate-to-hugs)[*View Source Project*](https://github.com/sdoran35/hate-to-hugs/tree/master/venv/lib/python3.6/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 24**

|  |  |  |
| --- | --- | --- |
| Project: *sdk-samples*   Author: *cradlepoint*   File: *authorizers.py*    [(license)](https://github.com/cradlepoint/sdk-samples)[*View Source Project*](https://github.com/cradlepoint/sdk-samples/tree/master/ftp_server/pyftpdlib/authorizers.py) | 6 votes | vote downvote up |

def get\_home\_dir(self, username):

"""Return the user's profile directory, the closest thing

to a user home directory we have on Windows.

"""

try:

sid = win32security.ConvertSidToStringSid(

win32security.LookupAccountName(None, username)[0])

except pywintypes.error as err:

raise AuthorizerError(err)

path = r"SOFTWARE\Microsoft\Windows NT" \

r"\CurrentVersion\ProfileList" + "\\" + sid

try:

key = winreg.OpenKey(winreg.HKEY\_LOCAL\_MACHINE, path)

except WindowsError:

raise AuthorizerError(

"No profile directory defined for user %s" % username)

value = winreg.QueryValueEx(key, "ProfileImagePath")[0]

home = win32api.ExpandEnvironmentStrings(value)

if not PY3 and not isinstance(home, unicode):

home = home.decode('utf8')

return home

**Example 25**

|  |  |  |
| --- | --- | --- |
| Project: *nsf2x*   Author: *adb014*   File: *mapiex.py*    [(license)](https://github.com/adb014/nsf2x)[*View Source Project*](https://github.com/adb014/nsf2x/tree/master/mapiex.py) | 6 votes | vote downvote up |

def CoCreateInstanceC2R (self, store, reg, clsid, iid) :

# Ugly code to find DLL in C2R version of COM object and get a COM

# object despite the fact that COM doesn't handle C2R

try:

# Get DLL to load from 2R register

aReg = winreg.ConnectRegistry(None, store)

aKey = winreg.OpenKey(aReg, reg, 0, winreg.KEY\_READ | winreg.KEY\_WOW64\_64KEY)

dummy\_n, IconvDLL, dummy\_t = winreg.EnumValue(aKey, 0)

winreg.CloseKey(aKey)

winreg.CloseKey(aReg)

# Create OLE object from DLL

IconvOLE = ctypes.OleDLL(IconvDLL)

# Get COM Instance from OLE

clsid\_class = uuid.UUID(str(clsid)).bytes\_le

iclassfactory = uuid.UUID(str(pythoncom.IID\_IClassFactory)).bytes\_le

com\_classfactory = ctypes.c\_long(0)

IconvOLE.DllGetClassObject(clsid\_class, iclassfactory, ctypes.byref(com\_classfactory))

MyFactory = pythoncom.ObjectFromAddress(com\_classfactory.value, pythoncom.IID\_IClassFactory)

return MyFactory.CreateInstance (None, str(iid))

except:

return None

**Example 26**

|  |  |  |
| --- | --- | --- |
| Project: *minihydra*   Author: *VillanCh*   File: *appdirs.py*    [(license)](https://github.com/VillanCh/minihydra)[*View Source Project*](https://github.com/VillanCh/minihydra/tree/master/py2env/lib/python2.7/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 27**

|  |  |  |
| --- | --- | --- |
| Project: *zenchmarks*   Author: *squeaky-pl*   File: *appdirs.py*    [(license)](https://github.com/squeaky-pl/zenchmarks)[*View Source Project*](https://github.com/squeaky-pl/zenchmarks/tree/master/vendor/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 28**

|  |  |  |
| --- | --- | --- |
| Project: *kbe\_server*   Author: *xiaohaoppy*   File: *installer.py*    [(license)](https://github.com/xiaohaoppy/kbe_server)[*View Source Project*](https://github.com/xiaohaoppy/kbe_server/tree/master/kbengine/kbe/tools/server/install/installer.py) | 6 votes | vote downvote up |

def getEnvironment(scope, name):

assert scope in ('user', 'system')

value = ''

if platform.system() == 'Windows':

root, subkey = getWindowsEnvironmentKey(scope)

key = winreg.OpenKey(root, subkey, 0, winreg.KEY\_READ)

try:

value, \_ = winreg.QueryValueEx(key, name)

except WindowsError:

value = ''

else:

if name.lower() == 'uid':

return str(os.geteuid())

if len(os\_user\_name) > 0:

ret, cret = syscommand('su -l %s -c \'echo ${%s}\'' % (os\_user\_name, name), True)

if len(ret) > 0:

value = ret[0].strip()

else:

value = os.environ.get(name, "")

return value

**Example 29**

|  |  |  |
| --- | --- | --- |
| Project: *sam-s-club-auctions*   Author: *sameer2800*   File: *appdirs.py*    [(license)](https://github.com/sameer2800/sam-s-club-auctions)[*View Source Project*](https://github.com/sameer2800/sam-s-club-auctions/tree/master/flask/lib/python3.6/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 30**

|  |  |  |
| --- | --- | --- |
| Project: *blog\_flask*   Author: *momantai*   File: *appdirs.py*    [(license)](https://github.com/momantai/blog_flask)[*View Source Project*](https://github.com/momantai/blog_flask/tree/master/lib/python3.5/site-packages/appdirs.py) | 6 votes | vote downvote up |

def \_get\_win\_folder\_from\_registry(csidl\_name):

"""This is a fallback technique at best. I'm not sure if using the

registry for this guarantees us the correct answer for all CSIDL\_\*

names.

"""

if PY3:

import winreg as \_winreg

else:

import \_winreg

shell\_folder\_name = {

"CSIDL\_APPDATA": "AppData",

"CSIDL\_COMMON\_APPDATA": "Common AppData",

"CSIDL\_LOCAL\_APPDATA": "Local AppData",

}[csidl\_name]

key = \_winreg.OpenKey(

\_winreg.HKEY\_CURRENT\_USER,

r"Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"

)

dir, type = \_winreg.QueryValueEx(key, shell\_folder\_name)

return dir

**Example 31**

|  |  |  |
| --- | --- | --- |
| Project: *my-first-blog*   Author: *AnkurBegining*   File: *utils.py*    [(license)](https://github.com/AnkurBegining/my-first-blog)[*View Source Project*](https://github.com/AnkurBegining/my-first-blog/tree/master/myenv/lib/python3.6/site-packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 32**

|  |  |  |
| --- | --- | --- |
| Project: *googletranslate.popclipext*   Author: *wizyoung*   File: *utils.py*    [(MIT License)](https://github.com/wizyoung/googletranslate.popclipext)[*View Source Project*](https://github.com/wizyoung/googletranslate.popclipext/tree/master/src/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 33**

|  |  |  |
| --- | --- | --- |
| Project: *pip-update-requirements*   Author: *alanhamlett*   File: *utils.py*    [(BSD 2-Clause "Simplified" License)](https://github.com/alanhamlett/pip-update-requirements)[*View Source Project*](https://github.com/alanhamlett/pip-update-requirements/tree/master/pur/packages/pip/_vendor/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 34**

|  |  |  |
| --- | --- | --- |
| Project: *paste2box*   Author: *rokups*   File: *settings.py*    [(GNU General Public License v3.0)](https://github.com/rokups/paste2box)[*View Source Project*](https://github.com/rokups/paste2box/tree/master/gui/settings.py) | 5 votes | vote downvote up |

def \_save\_autostart\_win(self, on):

with winreg.OpenKey(winreg.HKEY\_CURRENT\_USER, self.\_windows\_run\_reg\_key, 0, winreg.KEY\_ALL\_ACCESS) as key:

if on:

winreg.SetValueEx(key, const.APP\_NAME, 0, winreg.REG\_SZ, self.\_get\_executable\_path())

else:

try:

winreg.DeleteValue(key, const.APP\_NAME)

except OSError:

pass # key does not exist

**Example 35**

|  |  |  |
| --- | --- | --- |
| Project: *jira\_worklog\_scanner*   Author: *pgarneau*   File: *utils.py*    [(license)](https://github.com/pgarneau/jira_worklog_scanner)[*View Source Project*](https://github.com/pgarneau/jira_worklog_scanner/tree/master/jira_worklog_scanner/lib/python2.7/site-packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 36**

|  |  |  |
| --- | --- | --- |
| Project: *workflows.kyoyue*   Author: *wizyoung*   File: *utils.py*    [(MIT License)](https://github.com/wizyoung/workflows.kyoyue)[*View Source Project*](https://github.com/wizyoung/workflows.kyoyue/tree/master/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 37**

|  |  |  |
| --- | --- | --- |
| Project: *sublime-text-3-packages*   Author: *nickjj*   File: *img.py*    [(MIT License)](https://github.com/nickjj/sublime-text-3-packages)[*View Source Project*](https://github.com/nickjj/sublime-text-3-packages/tree/master/Packages/pygments/all/pygments/formatters/img.py) | 5 votes | vote downvote up |

def \_create\_win(self):

try:

key = \_winreg.OpenKey(

\_winreg.HKEY\_LOCAL\_MACHINE,

r'Software\Microsoft\Windows NT\CurrentVersion\Fonts')

except EnvironmentError:

try:

key = \_winreg.OpenKey(

\_winreg.HKEY\_LOCAL\_MACHINE,

r'Software\Microsoft\Windows\CurrentVersion\Fonts')

except EnvironmentError:

raise FontNotFound('Can\'t open Windows font registry key')

try:

path = self.\_lookup\_win(key, self.font\_name, STYLES['NORMAL'], True)

self.fonts['NORMAL'] = ImageFont.truetype(path, self.font\_size)

for style in ('ITALIC', 'BOLD', 'BOLDITALIC'):

path = self.\_lookup\_win(key, self.font\_name, STYLES[style])

if path:

self.fonts[style] = ImageFont.truetype(path, self.font\_size)

else:

if style == 'BOLDITALIC':

self.fonts[style] = self.fonts['BOLD']

else:

self.fonts[style] = self.fonts['NORMAL']

finally:

\_winreg.CloseKey(key)

**Example 38**

|  |  |  |
| --- | --- | --- |
| Project: *devsecops-example-helloworld*   Author: *boozallen*   File: *firefox\_binary.py*    [(license)](https://github.com/boozallen/devsecops-example-helloworld)[*View Source Project*](https://github.com/boozallen/devsecops-example-helloworld/tree/master/webapp/src/test/resources/lib/python2.6/site-packages/selenium/webdriver/firefox/firefox_binary.py) | 5 votes | vote downvote up |

def \_find\_exe\_in\_registry(self):

try:

from \_winreg import OpenKey, QueryValue, HKEY\_LOCAL\_MACHINE, HKEY\_CURRENT\_USER

except ImportError:

from winreg import OpenKey, QueryValue, HKEY\_LOCAL\_MACHINE, HKEY\_CURRENT\_USER

import shlex

keys = (

r"SOFTWARE\Classes\FirefoxHTML\shell\open\command",

r"SOFTWARE\Classes\Applications\firefox.exe\shell\open\command"

)

command = ""

for path in keys:

try:

key = OpenKey(HKEY\_LOCAL\_MACHINE, path)

command = QueryValue(key, "")

break

except OSError:

try:

key = OpenKey(HKEY\_CURRENT\_USER, path)

command = QueryValue(key, "")

break

except OSError:

pass

else:

return ""

if not command:

return ""

return shlex.split(command)[0]

**Example 39**

|  |  |  |
| --- | --- | --- |
| Project: *wow-addon-updater*   Author: *kuhnerdm*   File: *utils.py*    [(license)](https://github.com/kuhnerdm/wow-addon-updater)[*View Source Project*](https://github.com/kuhnerdm/wow-addon-updater/tree/master/packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 40**

|  |  |  |
| --- | --- | --- |
| Project: *bpy\_lambda*   Author: *bcongdon*   File: *ui.py*    [(license)](https://github.com/bcongdon/bpy_lambda)[*View Source Project*](https://github.com/bcongdon/bpy_lambda/tree/master/bpy_lambda/2.78/scripts/addons/render_povray/ui.py) | 5 votes | vote downvote up |

def locate\_docpath():

addon\_prefs = bpy.context.user\_preferences.addons[\_\_package\_\_].preferences

# Use the system preference if its set.

pov\_documents = addon\_prefs.docpath\_povray

if pov\_documents:

if os.path.exists(pov\_documents):

return pov\_documents

else:

print("User Preferences path to povray documents %r NOT FOUND, checking $PATH" % pov\_documents)

# Windows Only

if sys.platform[:3] == "win":

import winreg

try:

win\_reg\_key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

"Software\\POV-Ray\\v3.7\\Windows")

win\_docpath = winreg.QueryValueEx(win\_reg\_key, "DocPath")[0]

pov\_documents = os.path.join(win\_docpath, "Insert Menu")

if os.path.exists(pov\_documents):

return pov\_documents

except FileNotFoundError:

return""

# search the path all os's

pov\_documents\_default = "include"

os\_path\_ls = os.getenv("PATH").split(':') + [""]

for dir\_name in os\_path\_ls:

pov\_documents = os.path.join(dir\_name, pov\_documents\_default)

if os.path.exists(pov\_documents):

return pov\_documents

return ""

**Example 41**

|  |  |  |
| --- | --- | --- |
| Project: *infraview*   Author: *a-dekker*   File: *utils.py*    [(license)](https://github.com/a-dekker/infraview)[*View Source Project*](https://github.com/a-dekker/infraview/tree/master/python/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 42**

|  |  |  |
| --- | --- | --- |
| Project: *RPoint*   Author: *george17-meet*   File: *utils.py*    [(license)](https://github.com/george17-meet/RPoint)[*View Source Project*](https://github.com/george17-meet/RPoint/tree/master/gmaps/lib/python2.7/site-packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 43**

|  |  |  |
| --- | --- | --- |
| Project: *flasky*   Author: *RoseOu*   File: *firefox\_binary.py*    [(license)](https://github.com/RoseOu/flasky)[*View Source Project*](https://github.com/RoseOu/flasky/tree/master/venv/lib/python2.7/site-packages/selenium/webdriver/firefox/firefox_binary.py) | 5 votes | vote downvote up |

def \_find\_exe\_in\_registry(self):

try:

from \_winreg import OpenKey, QueryValue, HKEY\_LOCAL\_MACHINE, HKEY\_CURRENT\_USER

except ImportError:

from winreg import OpenKey, QueryValue, HKEY\_LOCAL\_MACHINE, HKEY\_CURRENT\_USER

import shlex

keys = (

r"SOFTWARE\Classes\FirefoxHTML\shell\open\command",

r"SOFTWARE\Classes\Applications\firefox.exe\shell\open\command"

)

command = ""

for path in keys:

try:

key = OpenKey(HKEY\_LOCAL\_MACHINE, path)

command = QueryValue(key, "")

break

except OSError:

try:

key = OpenKey(HKEY\_CURRENT\_USER, path)

command = QueryValue(key, "")

break

except OSError:

pass

else:

return ""

if not command:

return ""

return shlex.split(command)[0]

**Example 44**

|  |  |  |
| --- | --- | --- |
| Project: *macos-st-packages*   Author: *zce*   File: *img.py*    [(license)](https://github.com/zce/macos-st-packages)[*View Source Project*](https://github.com/zce/macos-st-packages/tree/master/Packages/pygments/all/pygments/formatters/img.py) | 5 votes | vote downvote up |

def \_create\_win(self):

try:

key = \_winreg.OpenKey(

\_winreg.HKEY\_LOCAL\_MACHINE,

r'Software\Microsoft\Windows NT\CurrentVersion\Fonts')

except EnvironmentError:

try:

key = \_winreg.OpenKey(

\_winreg.HKEY\_LOCAL\_MACHINE,

r'Software\Microsoft\Windows\CurrentVersion\Fonts')

except EnvironmentError:

raise FontNotFound('Can\'t open Windows font registry key')

try:

path = self.\_lookup\_win(key, self.font\_name, STYLES['NORMAL'], True)

self.fonts['NORMAL'] = ImageFont.truetype(path, self.font\_size)

for style in ('ITALIC', 'BOLD', 'BOLDITALIC'):

path = self.\_lookup\_win(key, self.font\_name, STYLES[style])

if path:

self.fonts[style] = ImageFont.truetype(path, self.font\_size)

else:

if style == 'BOLDITALIC':

self.fonts[style] = self.fonts['BOLD']

else:

self.fonts[style] = self.fonts['NORMAL']

finally:

\_winreg.CloseKey(key)

**Example 45**

|  |  |  |
| --- | --- | --- |
| Project: *StarCraft-Casting-Tool*   Author: *teampheenix*   File: *config.py*    [(license)](https://github.com/teampheenix/StarCraft-Casting-Tool)[*View Source Project*](https://github.com/teampheenix/StarCraft-Casting-Tool/tree/master/scctool/settings/config.py) | 5 votes | vote downvote up |

def findTesserAct(default="C:\\Program Files (x86)\\Tesseract-OCR\\tesseract.exe"):

"""Search for Tesseract exceutable via registry."""

if(not windows):

return default

try:

import winreg

key = winreg.OpenKey(winreg.HKEY\_LOCAL\_MACHINE,

"SOFTWARE\\WOW6432Node\\Tesseract-OCR")

return winreg.QueryValueEx(key, "Path")[0] + '\\tesseract.exe'

except:

return default

**Example 46**

|  |  |  |
| --- | --- | --- |
| Project: *pyVirtualize*   Author: *rocky1109*   File: *klasses.py*    [(license)](https://github.com/rocky1109/pyVirtualize)[*View Source Project*](https://github.com/rocky1109/pyVirtualize/tree/master/src/pyVirtualize/utils/klasses.py) | 5 votes | vote downvote up |

def getenv(self, name):

key = winreg.OpenKey(self.root, self.subkey, 0, winreg.KEY\_READ)

try:

value, \_ = winreg.QueryValueEx(key, name)

except WindowsError:

value = ''

winreg.CloseKey(key)

return value

**Example 47**

|  |  |  |
| --- | --- | --- |
| Project: *pyVirtualize*   Author: *rocky1109*   File: *klasses.py*    [(license)](https://github.com/rocky1109/pyVirtualize)[*View Source Project*](https://github.com/rocky1109/pyVirtualize/tree/master/src/pyVirtualize/utils/klasses.py) | 5 votes | vote downvote up |

def setenv(self, name, value):

# Note: for 'system' scope, you must run this as Administrator

key = winreg.OpenKey(self.root, self.subkey, 0, winreg.KEY\_ALL\_ACCESS)

winreg.SetValueEx(key, name, 0, winreg.REG\_EXPAND\_SZ, value)

winreg.CloseKey(key)

# For some strange reason, calling SendMessage from the current process

# doesn't propagate environment changes at all.

# TODO: handle CalledProcessError (for assert)

check\_call('''\

"%s" -c "import win32api, win32con; assert win32api.SendMessage(win32con.HWND\_BROADCAST, win32con.WM\_SETTINGCHANGE, 0, 'Environment')"''' % sys.executable)

**Example 48**

|  |  |  |
| --- | --- | --- |
| Project: *flickr\_downloader*   Author: *Denisolt*   File: *utils.py*    [(license)](https://github.com/Denisolt/flickr_downloader)[*View Source Project*](https://github.com/Denisolt/flickr_downloader/tree/master/local/lib/python2.7/site-packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 49**

|  |  |  |
| --- | --- | --- |
| Project: *SHAREOpenRefineWkshop*   Author: *cmh2166*   File: *utils.py*    [(license)](https://github.com/cmh2166/SHAREOpenRefineWkshop)[*View Source Project*](https://github.com/cmh2166/SHAREOpenRefineWkshop/tree/master/Data/venv/lib/python3.5/site-packages/requests/utils.py) | 5 votes | vote downvote up |

def proxy\_bypass\_registry(host):

if is\_py3:

import winreg

else:

import \_winreg as winreg

try:

internetSettings = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,

r'Software\Microsoft\Windows\CurrentVersion\Internet Settings')

proxyEnable = winreg.QueryValueEx(internetSettings,

'ProxyEnable')[0]

proxyOverride = winreg.QueryValueEx(internetSettings,

'ProxyOverride')[0]

except OSError:

return False

if not proxyEnable or not proxyOverride:

return False

# make a check value list from the registry entry: replace the

# '<local>' string by the localhost entry and the corresponding

# canonical entry.

proxyOverride = proxyOverride.split(';')

# now check if we match one of the registry values.

for test in proxyOverride:

if test == '<local>':

if '.' not in host:

return True

test = test.replace(".", r"\.") # mask dots

test = test.replace("\*", r".\*") # change glob sequence

test = test.replace("?", r".") # change glob char

if re.match(test, host, re.I):

return True

return False

**Example 50**

|  |  |  |
| --- | --- | --- |
| Project: *nimp*   Author: *dontnod*   File: *build.py*    [(license)](https://github.com/dontnod/nimp)[*View Source Project*](https://github.com/dontnod/nimp/tree/master/nimp/build.py) | 5 votes | vote downvote up |

def \_find\_devenv\_path(vs\_version):

devenv\_path = None

# First try the registry, because the environment variable is unreliable

# (case of Visual Studio installed on a different drive; it still sets

# the envvar to point to C:\Program Files even if devenv.com is on D:\)

#pylint: disable=import-error

from winreg import OpenKey, QueryValue, HKEY\_LOCAL\_MACHINE

key\_path = 'SOFTWARE\\Classes\\VisualStudio.accessor.' + vs\_version + '.0\\shell\\Open'

try:

with OpenKey(HKEY\_LOCAL\_MACHINE, key\_path) as key:

cmdline = QueryValue(key, 'Command')

if cmdline[:1] == '"':

cmdline = cmdline.split('"')[1]

elif ' ' in cmdline:

cmdline = cmdline.split(' ')[0]

devenv\_path = cmdline.replace('devenv.exe', 'devenv.com')

#pylint: disable=broad-except

except Exception:

pass

# If the registry key is unhelpful, try the environment variable

if not devenv\_path:

vstools\_path = os.getenv('VS' + vs\_version + '0COMNTOOLS')

if vstools\_path is not None:

# Sanitize this because os.path.join sometimes gets confused

if vstools\_path[-1] in [ '/', '\\' ]:

vstools\_path = vstools\_path[:-1]

devenv\_path = os.path.join(vstools\_path, '../../Common7/IDE/devenv.com')

if not devenv\_path or not os.path.exists(devenv\_path):

return None

logging.info("Found Visual Studio at %s", devenv\_path)

return devenv\_path