
PyWin32ctypes Documentation

Release 0.0.2.dev1

David Cournapeau, Ioannis Tziakos

February 22, 2016

A reimplementation of pywin32 that is pure python. The default behaviour will try to use cffi ($\geq 1.3.0$), if available, and fall back to using ctypes. Please note that there is no need to have a compiler available on installation or at runtime.

Usage

Example:

```
# Equivalent to 'import win32api' from pywin32.  
from win32ctypes.pywin32 import win32api  
  
win32api.LoadLibraryEx(sys.executable, 0, win32api.LOAD_LIBRARY_AS_DATAFILE)
```

Note: Currently pywin32ctypes implements only a very small subset of pywin32, for internal needs at Enthought. We do welcome additional features and PRs, though.

Development setup

The following should be good enough:

```
pip install -r dev_requirements.txt
python install -e
```

Note:

- While pywin32-ctypes should regularly be tested on windows, you can also develop/test on unix by using wine (see travis-ci configuration to set it up).
-

Reference

3.1 PyWin32 Compatibility Layer

The win32ctypes library provides a compatibility layer with PyWin32. We attempt to keep the signature of the Functions and their behaviour as close as possible to PyWin32.

<code>win32ctypes.pywin32.win32api</code>	A module, encapsulating the Windows Win32 API.
<code>win32ctypes.pywin32.win32cred</code>	Interface to credentials management functions.
<code>win32ctypes.pywin32.pywintypes</code>	A module which supports common Windows types.

3.1.1 win32api

A module, encapsulating the Windows Win32 API.

Functions

<code>EnumResourceLanguages(hModule, lpType, lpName)</code>	List languages of a resource module.
<code>EnumResourceNames(hModule, resType)</code>	Enumerates all the resources of the specified type within a module.
<code>EnumResourceTypes(hModule)</code>	Enumerates resource types within a module.
<code>FreeLibrary(hModule)</code>	Free the loaded dynamic-link library (DLL) module.
<code>GetTickCount()</code>	
<code>LoadLibraryEx(fileName, handle, flags)</code>	Loads the specified DLL, and returns the handle.
<code>LoadResource(hModule, type, name[, language])</code>	Find and Load a resource component.

`win32ctypes.pywin32.win32api.EnumResourceLanguages(hModule, lpType, lpName)`
List languages of a resource module.

Parameters **hModule** (*handle*) – Handle to the resource module.

lpType [str][int] The type of resource to enumerate. If `lpType` is a string starting with '#' is should be followed by the decimal number that define the integer resource type identifier.

lpName [str][int] The name of resource to enumerate. If `lpType` is a string starting with '#' is should be

followed by the decimal number that define the integer resource type identifier.

Returns `resource_languages` (*list*) – List of the resource language ids.

See also

- [EnumResourceLanguages MSDN reference](#)
- [Predefined resource types](#)
- [Predefined resource language ids](#)

`win32ctypes.pywin32.win32api.EnumResourceNames` (*hModule*, *resType*)

Enumerates all the resources of the specified type within a module.

Parameters `hModule` (*handle*) – The handle to the module.

resType [*str*][*int*] The type of resource to enumerate. If `resType` is a string starting with '#' is should be followed by the decimal number that define the integer resource type identifier.

Returns `resource_names` (*list*) – The list of resource names (unicode strings) of the specific resource type in the module.

See also

- [EnumResourceNames MSDN reference](#)
- [Predefined resource types](#)

`win32ctypes.pywin32.win32api.EnumResourceTypes` (*hModule*)

Enumerates resource types within a module.

Parameters `hModule` (*handle*) – The handle to the module.

Returns `resource_types` (*list*) – The list of resource types in the module.

See also

- [EnumResourceTypes MSDN reference](#)

`win32ctypes.pywin32.win32api.FreeLibrary` (*hModule*)

Free the loaded dynamic-link library (DLL) module.

If necessary, decrements its reference count.

Parameters `hModule` – The handle to the library as returned by the `LoadLibrary` function.

`win32ctypes.pywin32.win32api.GetTickCount` ()

`win32ctypes.pywin32.win32api.LoadLibraryEx` (*fileName*, *handle*, *flags*)

Loads the specified DLL, and returns the handle.

Parameters

- **fileName** (*unicode*) – The filename of the module to load.
- **handle** – Reserved, always zero.

- **flags** – The action to be taken when loading the module.

Returns **hModule** – The handle of the loaded module

See also

- [LoadLibraryEx MSDN reference](#)

`win32ctypes.pywin32.win32api.LoadResource(hModule, type, name, language=0)`

Find and Load a resource component.

Parameters **handle** – The handle of the module containing the resource. Use None for current process executable.

type [str][int] The type of resource to load.

name : The name or Id of the resource to load.

language [int] Language to use, default is LANG_NEUTRAL.

Returns **hModule** – Handle of the loaded source.

See also

- [FindResourceEx MSDN reference](#)
- [SizeofResource MSDN reference](#)
- [LoadResource MSDN reference](#)
- [LockResource MSDN reference](#)
- [Predefined resource types](#)
- [Predefined resource language ids](#)

3.1.2 win32cred

Interface to credentials management functions.

Functions

<code>CredDelete(TargetName, Type[, Flags])</code>	Remove the given target name from the stored credentials.
<code>CredRead(TargetName, Type[, Flags])</code>	Retrieves a stored credential.
<code>CredWrite(Credential[, Flags])</code>	Creates or updates a stored credential.
<code>pywin32error(*args, **kwds)</code>	

`win32ctypes.pywin32.win32cred.CredDelete(TargetName, Type, Flags=0)`

Remove the given target name from the stored credentials.

Parameters

- **TargetName** (*unicode*) – The target name to fetch from the keyring.
- **Type** (*int*) – One of the CRED_TYPE_* constants.
- **Flags** (*int*) – Reserved, always use 0.

`win32ctypes.pywin32.win32cred.CredRead(TargetName, Type, Flags=0)`
Retrieves a stored credential.

Parameters

- **TargetName** (*unicode*) – The target name to fetch from the keyring.
- **Type** (*int*) – One of the CRED_TYPE_* constants.
- **Flags** (*int*) – Reserved, always use 0.

Returns

credentials (*dict*) – A dictionary containing the following:

- **UserName**: the retrieved username
- **CredentialBlob**: the password (as an utf-16 encoded 'string')

None if the target name was not found.

`win32ctypes.pywin32.win32cred.CredWrite(Credential, Flags=0)`
Creates or updates a stored credential.

Parameters

- **Credential** (*dict*) – Parameters to be passed to win32 API CredWrite/
- **Flags** (*int*) – Always pass CRED_PRESERVE_CREDENTIAL_BLOB (i.e. 0).

Returns

credentials (*dict*) – A dictionary containing the following:

- **Type**: the type of credential (see MSDN)
- **TargetName**: the target to use (string)
- **Persist**: see MSDN
- **UserName**: the retrieved username
- **CredentialBlob**: the password (as a *string*, not an encoded binary stream - this function takes care of the encoding).
- **Comment**: a string

`win32ctypes.pywin32.win32cred.pywin32error(*args, **kws)`

3.1.3 pywintypes

A module which supports common Windows types.

Functions

`pywin32error(*args, **kws)`

```
win32ctypes.pywin32.pywintypes.pywin32error(*args, **kws)
```

Exceptions

```
error(*args, **kw)
```

```
class win32ctypes.pywin32.pywintypes.error(*args, **kw)
```

Change Log

4.1 Current development version

- Add documentation and setup automatic builds in ReadTheDocs (#3, #36).
- Add cffi backend to be used when available (#31).
- Fix issue where EnumResourceTypes and EnumResourceNames would only return ints (#21, #30).
- Restructure package layout to split core wrapping modules from pywin32 emulation (#15, #17).

4.2 Version 0.0.1

7/04/2014

- Python 2.6 support (#13)
- Python 3 support (#12)
- Basic maintenance work (#11, #7)
- Fix error raising to be pywin32 compatible (#8)
- Package rename mini_pywin32 -> pywin32-ctypes
- Add travis-ci integration using wine! (#2)
- Support basic library and resource loading (#1)
- mini_pywin32 is born

LICENSE

This software is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative.

Copyright (c) 2014, Enthought, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Enthought, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Indices and tables

- `genindex`
- `modindex`
- `search`

W

`win32ctypes.pywin32.pywintypes`, [10](#)
`win32ctypes.pywin32.win32api`, [7](#)
`win32ctypes.pywin32.win32cred`, [9](#)

C

CredDelete() (in module win32ctypes.pywin32.win32cred), 9

CredRead() (in module win32ctypes.pywin32.win32cred), 10

CredWrite() (in module win32ctypes.pywin32.win32cred), 10

E

EnumResourceLanguages() (in module win32ctypes.pywin32.win32api), 7

EnumResourceNames() (in module win32ctypes.pywin32.win32api), 8

EnumResourceTypes() (in module win32ctypes.pywin32.win32api), 8

error (class in win32ctypes.pywin32.pywintypes), 11

F

FreeLibrary() (in module win32ctypes.pywin32.win32api), 8

G

GetTickCount() (in module win32ctypes.pywin32.win32api), 8

L

LoadLibraryEx() (in module win32ctypes.pywin32.win32api), 8

LoadResource() (in module win32ctypes.pywin32.win32api), 9

P

pywin32error() (in module win32ctypes.pywin32.pywintypes), 11

pywin32error() (in module win32ctypes.pywin32.win32cred), 10

W

win32ctypes.pywin32.pywintypes (module), 10

win32ctypes.pywin32.win32api (module), 7