
win-nic Documentation

Release 2.0.1

Tyler N. Thieding

Jun 19, 2019

Contents

1	Quick Start	3
1.1	About	3
1.2	API Reference	4
1.3	Installation	6
1.4	Release Notes	6

Interface with network interface cards (NICs) on Windows-based computers. This package heavily wraps the Windows management instrumentation command-line (WMIC) and the netsh command-line utility via subprocess calls. No C dependencies or building from source files, just a lightweight and straightforward wrapper of utilities built into your Windows system.

CHAPTER 1

Quick Start

First, obtain a NIC instance via the `NetworkAdapters` class. To do this, instantiate `NetworkAdapters` and specify the desired NIC. Specify the target NIC by registry index, name, or connection ID (control panel name):

```
>>> from win_nic import NetworkAdapters
>>> this_pc_nics = NetworkAdapters()
>>> ethernet_nic = this_pc_nics.get_nic(connection_id="Local Area Connection 1")
```

Now, interface with the NIC instance as needed by getting attributes or calling methods:

```
>>> ethernet_nic.property.ip_addresses
['192.168.0.2']
>>> ethernet_nic.net_connection_status
<NicNetConnectionStatus.CONNECTED: 2>
>>> ethernet_nic.set_static_address('192.168.0.3', '255.255.255.0', '192.168.0.1')
0
```

1.1 About

1.1.1 Contributors

- Tyler N. Thieding (Primary Author)

1.1.2 Development

Repository <https://gitlab.com/TNThieding/win-nic>

Backlog <https://trello.com/b/SYbiCLBg/python-package-kanban-board-win-nic>

1.1.3 License

Copyright 2019 Tyler N. Thieding

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2 API Reference

- *Classes*
- *NIC Instance Attributes*
- *NIC Instance Methods*

1.2.1 Classes

- `modindex`
- `win_nic.NetworkAdapters`
- `win_nic.Nic`

1.2.2 NIC Instance Attributes

Refer to Microsoft Developer Network's pages on the `Win32_NetworkAdapter` and `Win32_NetworkAdapterConfiguration` classes for further information about the attributes listed below.

NIC Instance Attribute	Windows Class and Property	Python Type	Access
adapter_type	Win32_NetworkAdapter.AdapterType	NicAdapterType	Get
availability	Win32_NetworkAdapter.Availability	NicAvailability	Get
caption	Win32_NetworkAdapter.Caption	str	Get
config_manager_error_code	Win32_NetworkAdapter.ConfigManagerErrorCode	NicConfigManager- ErrorCode	Get
config_manager_user_config	Win32_NetworkAdapter.ConfigManagerUserConfig	bool	Get
description	Win32_NetworkAdapter.Description	str	Get
device_id	Win32_NetworkAdapter.DeviceID	str	Get
enabled_ctrl_panel	Custom Attribute (Is NIC Enabled in Control Panel?)	bool	Get
error_cleared	Win32_NetworkAdapter.ErrorCleared	bool	Get
error_description	Win32_NetworkAdapter.ErrorDescription	str	Get
guid	Win32_NetworkAdapter.GUID	str	Get
index	Win32_NetworkAdapter.Index	int	Get
installed	Win32_NetworkAdapter.Installed	bool	Get
interface_index	Win32_NetworkAdapter.InterfaceIndex	int	Get
ip_addresses	Win32_NetworkAdapterConfiguration.IPAddress	list of str	Get
last_error_code	Win32_NetworkAdapter.LastErrorCode	int	Get
mac_address	Win32_NetworkAdapter.MACAddress	str	Get
manufacturer	Win32_NetworkAdapter.Manufacturer	str	Get
name	Win32_NetworkAdapter.Name	str	Get
net_connection_id	Win32_NetworkAdapter.NetConnectionID	str	Get
net_connection_status	Win32_NetworkAdapter.NetConnectionStatus	NicNetConnectionStatus	Get
physical_adapter	Win32_NetworkAdapter.PhysicalAdapter	bool	Get
pnnp_device_id	Win32_NetworkAdapter.PNPDeviceID	str	Get
power_management_supported	Win32_NetworkAdapter.PowerManagementSupported	bool	Get
product_name	Win32_NetworkAdapter.ProductName	str	Get
service_name	Win32_NetworkAdapter.ServiceName	str	Get
speed	Win32_NetworkAdapter.Speed	int	Get

1.2.3 NIC Instance Methods

Refer to Microsoft Developer Network's pages on the [Win32_NetworkAdapter](#) and [Win32_NetworkAdapterConfiguration](#) classes for further information about the methods listed below.

NIC Instance Method	Windows Class and Method	Admin Access Required?	Parameters (In Order)	Returns
add_dns_server	Custom netsh Method (Add DNS Server Entry)	Yes	dns_server (str): DNS Server Address	Return Status Code (int)
disable	Win32_NetworkAdapter.Disable	Yes	None	Windows Error Code (int)
enable	Win32_NetworkAdapter.Enable	Yes	None	Windows Error Code (int)
set_static_address	Custom netsh Method (Set Static IP Address Configuration)	Yes	ip_addr (str): Static IP Address subnet_mask (str): Static Subnet Mask gateway (str): Static Default Gateway	Return Status Code (int)
use_dhcp	Custom netsh Method (Use DHCP to Obtain IP Address)	Yes	None	Return Status Code (int)

Tip: Decode a Windows error code by running `net helpmsg #`, where # is the integer return value.

Tip: When using a method requiring administrative privileges, run the script from an administrative command prompt by navigating to a command prompt shortcut, right clicking it, and choosing “Run as administrator” from the menu.

1.3 Installation

1.3.1 Requirements

- Python 3.6+
- Windows Vista or Later

1.3.2 Installation Steps

```
pip install win-nic
```

1.4 Release Notes

1.4.1 [2.0.1] Improve under-the-hood attribute access architecture. (2019-06-18)

Previously, the package supported attribute access with object-oriented design hand-crafted `@property` directives. Now, the package leverages Python built-in methods (e.g. `__getattr__`) to facilitate attribute access in a more maintainable and Pythonic manner.

1.4.2 [2.0.0] Remove `property` and `method` members and drop Python 2 support. (2019-06-16)

Previously, `Nic` instances housed accessors to properties and methods in the `property` and `method` attributes. Now, both properties and methods exist in the `Nic` namespace.

Remove legacy Python 2 syntax from code.

This release includes the following under-the-hood changes:

- Migrate repository from GitHub to GitLab (including CI/CD).
- Pylint cleanup regarding Python 3 syntax.

1.4.3 Prior Changes

- [1.0.1] Fix ReadTheDocs settings, make `enum34` a dependency. (2018-07-07)
- [1.0.0] Initial release. (2018-07-06)