

<https://www.halvorsen.blog>



# OPC UA in Python

Hans-Petter Halvorsen

# Contents

- Introduction
- OPC UA Server Simulator from Integration Objects
- Python OPC UA Client Examples
- OPC UA Python Client GUI
- Python OPC UA Server Examples



# Introduction

# Introduction

- This Tutorial will demonstrate OPC UA in Python with some basic Code Examples using the “**opcua**” Python package
- This Tutorial will use the **Thonny** Python Editor and the **Spyder** Python Editor, but other Python Editors can of course also be used
- This Tutorial will use the **OPC UA Server Simulator** from Integration Objects (free OPC UA server for Test and Demonstration purposes), but other OPC UA Servers can of course also be used
- We will create both OPC UA Clients and an OPC UA Server using Python and the “**opcua**” Python package

# Software

- **Thonny** Python Editor  
<https://thonny.org>
- **Spyder** Python Editor. You can download it separately or part of Anaconda Python Distribution  
<https://www.spyder-ide.org>  
<https://www.anaconda.com>
- **opcua** Python package/library  
<https://pypi.org/project/opcua/>
- **OPC UA Server Simulator:**  
<https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-server-simulator/>

# Installation using Thonny

Manage packages for C:\Users\hansha\AppData\Local\Programs\Thonny\python.exe

opcua Search on PyPI

<INSTALL>  
adafruit-board-toolkit  
astroid  
asttokens  
bcrypt  
bitstring  
cffi  
colorama  
contourpy  
cryptography  
cyclcr  
deprecation  
dill  
dnspython  
docutils  
ecdsa  
esptool  
fonttools  
guizero  
isort  
jedi  
kiwisolver  
lazy-object-proxy

## Search results

[opcua](#)  
Pure Python OPC-UA client and server library

[opcua-webclient](#)  
A OPCUA web client, implemented by Python.

[opcua-widgets](#)  
OPC-UA Widgets

[opcua-modeler](#)  
OPC-UA Address Space Modeler

[formic-opcua](#)  
<No description>

[opcua-client](#)  
OPC-UA Client GUI

[opcua-tools](#)

isort  
jedi  
kiwisolver  
lazy-object-proxy  
lxml  
matplotlib  
mccabe  
mypy  
mypy-extensions  
nidaqmx  
numpy  
**opcua**  
packaging  
paho-mqtt  
paramiko  
parso  
pillow  
pip  
platformdirs  
pyparser  
pylint  
pymongo  
pynacl

Manage packages for C:\Users\hansha\AppData\Local\Programs\Thonny\python.exe

opcua Search on PyPI

## opcua

**Installed version:** 0.98.13  
**Installed to:** C:\Users\hansha\AppData\Roaming\Python\Python310\site-packages

**Latest stable version:** 0.98.13  
**Summary:** Pure Python OPC-UA client and server library  
**Author:** Olivier Roulet-Dubonnet  
**Homepage:** <http://freeopcua.github.io/>  
**PyPI page:** <https://pypi.org/project/opcua/>

Upgrade Uninstall ... Close

# References

- PyPi:  
<https://pypi.org/project/opcua/>
- Python OPC-UA Documentation:  
<https://python-opcua.readthedocs.io/en/latest/>
- GitHub:  
<https://github.com/FreeOpcUa/python-opcua/tree/master/examples>
- GitHub Client GUI:  
<https://github.com/FreeOpcUa/opcua-client-gui>



# OPC UA Simulation Server



# OPC UA Server Simulator

- “OPC UA Server Simulator” from Integration Objects is a free OPC UA Server
- You can use it for Test and Demonstration purposes.
- It runs 48 hours before you need to restart it
- Users can configure their own OPC Tags via CSV files
- Download:  
<https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-server-simulator/>

# OPC UA Server Simulator

The screenshot shows the OPC UA Server Simulator application window. The title bar reads "OPC UA Server Simulator". The menu bar includes "File", "Settings", and "Help". A text field for "Server Endpoints URLs" contains the address "opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator".

The "Sessions" tab is active, displaying an empty table with the following headers:

SessionId	Name	User	Last Contact
-----------	------	------	--------------

The "Subscriptions" tab is also visible, displaying an empty table with the following headers:

SubscriptionId	Publishing Interval	Item Count	Seq No
----------------	---------------------	------------	--------

The status bar at the bottom provides the following information:

**Status:** Running    **Current Time:** 11:01:11    **Sessions:** 0    **Subscriptions:** 0    **Items:** 0

# OPC UA Client

- Integration Objects also offer a free OPC UA Client
- You can use it for Test and Demonstration purposes
- Download:  
<https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-client/>

# OPC UA Client

The screenshot displays the OPC UA Client application window titled "Integration Objects' OPC UA Client". The interface includes a ribbon menu with tabs for Home, File, Session, Configuration, Help, and Certificate Manager. The main workspace is divided into several panes: Sessions, Address Space, Forward, and Data View. A "Connection Settings" dialog box is open in the center, allowing configuration of session parameters. The dialog is set to "Session0" and "opc.tcp://xps15hph:62640/IntegrationObjects/!". The "Transport Protocol" is set to "Opc.tcp", "Message Encoding" to "Binary", "Security Mode" to "None", and "Security Policy" to "None". "User Authentication Mode" is set to "Anonymous".

**Connection Settings**

Session Information  
Session Name:

Server Information  
Endpoint Uri:

Transport Protocol  
 Opc.tcp  
 Https

Message Encoding  
 Binary  
 Xml

Security Mode  
 None  
 Sign  
 Sign\_Encrypt

Security Policy  
 None  
 Basic128RSA15  
 Basic256  
 Basic256Sha256

User Authentication Mode  
 Anonymous  UserName  Certificate

Certificate (.pfx):

Password:

**Message Log**

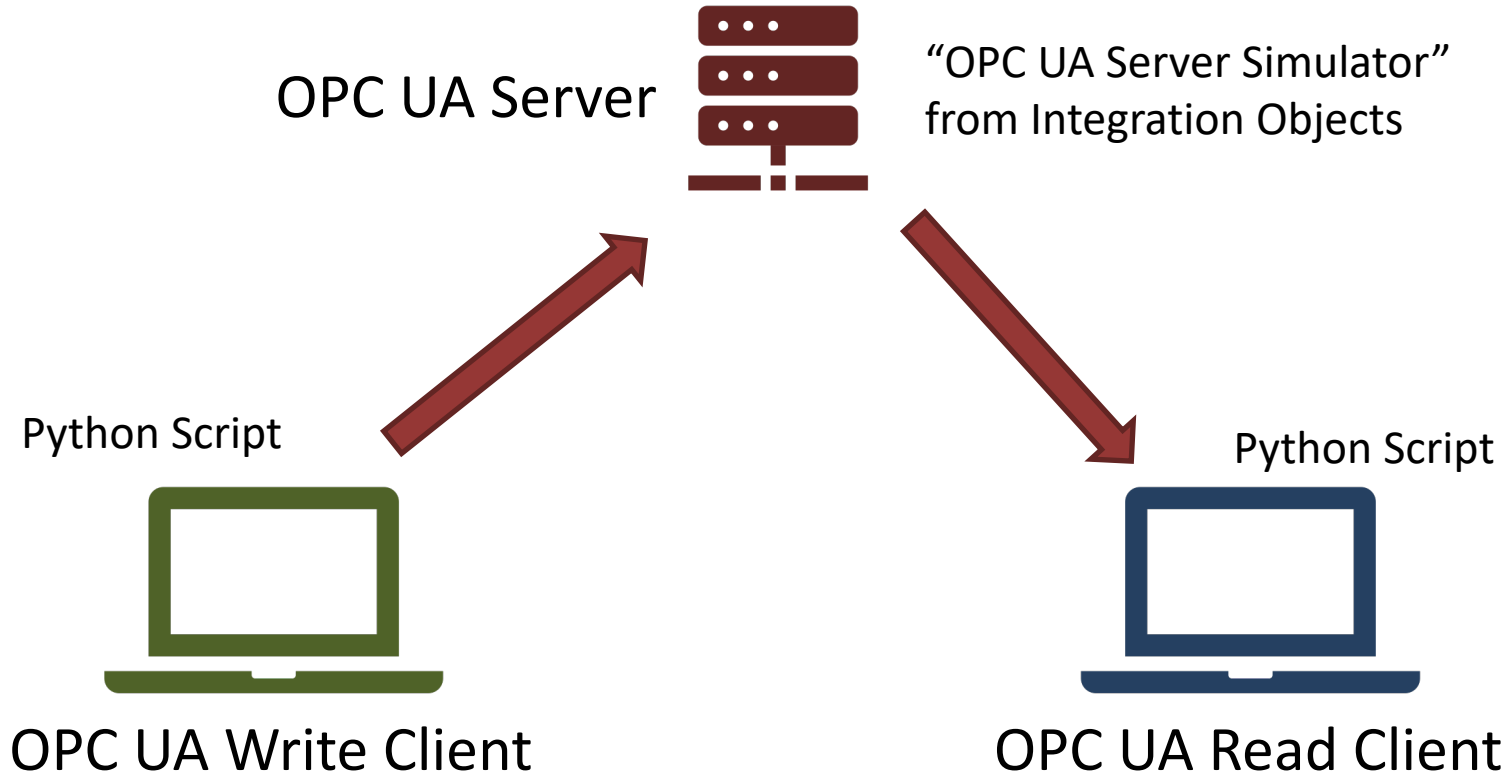
Message Type	Timestamp	Message
[Control]	2022-02-08 13:05:06	Disconnecting from session
[Control]	2022-02-08 13:03:09	Read operation of the variab
[Control]	2022-02-08 13:01:03	A session "Session0" with the

3 Messages

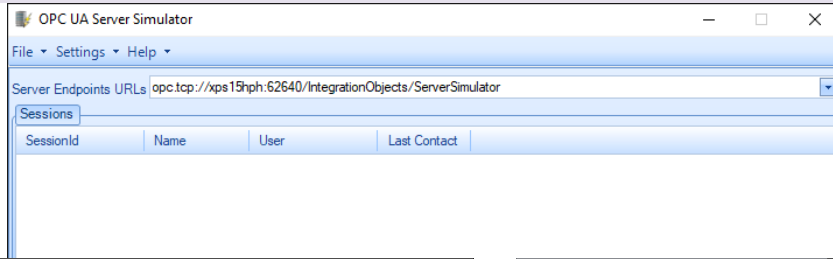


# Python OPC UA Client Examples

# Python OPC UA Client Examples



# Python OPC UA Client Examples



```
Thonny - C:\Users\hansha\OneDrive\Documents\Python\Python Programming\Code Examples\OPC\opcua\Integration Objects Examples\opcua_client_write_ex.py @ 11:17
File Edit View Run Tools Help
opcua_client_write_ex.py | opcua_client_read_ex.py
1 from opcua import Client
2
3 #OPC UA Server Simulator from Integration Objects
4 url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
5 nodeId = "ns=2;s=Tag7"
6
7 client = Client(url)
8 client.connect()
9
10 node = client.get_node(nodeId)
11 value = float(22)
12 node.set_data_value(value)
13
14 client.disconnect()
Shell x
>>> %Run opcua_client_write_ex.py
>>>
```

Local Python 3 • Thonny's Python

```
Thonny - C:\Users\hansha\OneDrive\Documents\Python\Python Programming\Code Examples\OPC\opcua\Integration Objects Examples\opcua_client_read_ex.py @ 10:31
File Edit View Run Tools Help
opcua_client_write_ex.py | opcua_client_read_ex.py
1 from opcua import Client
2
3 #OPC UA Server Simulator from Integration Objects
4 url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
5 nodeId = "ns=2;s=Tag7"
6
7 client = Client(url)
8 client.connect()
9
10 node = client.get_node(nodeId)
11 value = node.get_value()
12 print(value)
13
14 client.disconnect()
Shell x
>>> %Run opcua_client_read_ex.py
22.0
>>>
```

Local Python 3 • Thonny's Python

# OPC UA Client Write Example

```
from opcua import Client

#OPC UA Server Simulator from Integration Objects
url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
nodeId = "ns=2;s=Tag7"

client = Client(url)
client.connect()

node = client.get_node(nodeId)
value = float(20)
node.set_data_value(value)

client.disconnect()
```



# OPC UA Client Read Example

```
from opcua import Client

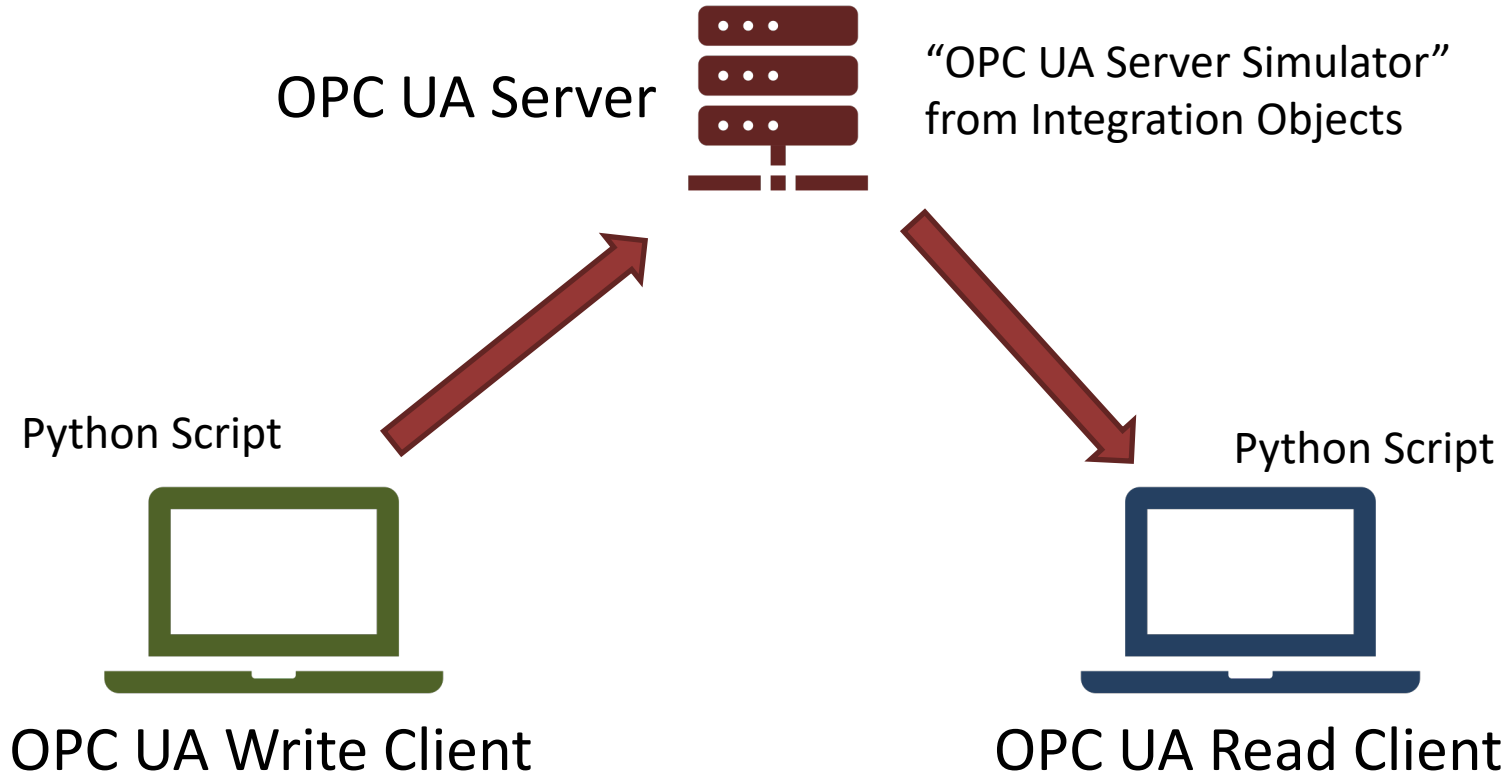
#OPC UA Server Simulator from Integration Objects
url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
nodeId = "ns=2;s=Tag7"

client = Client(url)
client.connect()

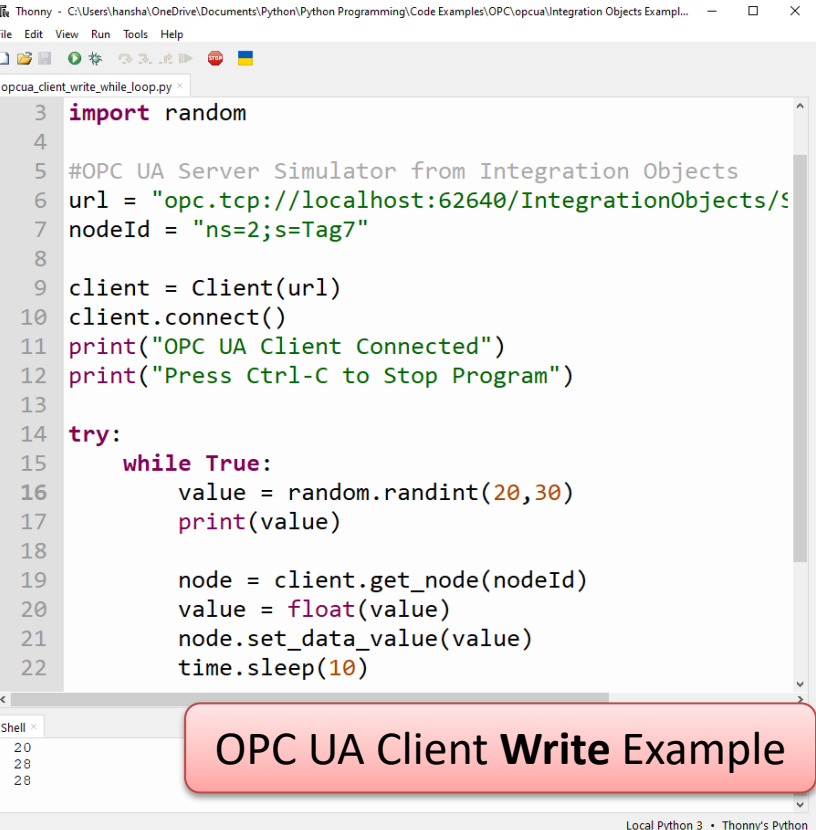
node = client.get_node(nodeId)
value = node.get_value()
print(value)

client.disconnect()
```

# Improved Example with While Loop

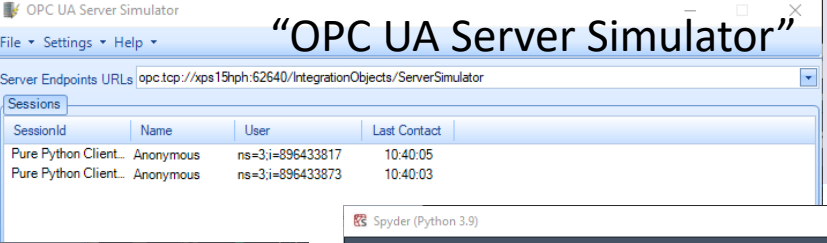


# Thonny Python Editor



```
3 import random
4
5 #OPC UA Server Simulator from Integration Objects
6 url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
7 nodeId = "ns=2;s=Tag7"
8
9 client = Client(url)
10 client.connect()
11 print("OPC UA Client Connected")
12 print("Press Ctrl-C to Stop Program")
13
14 try:
15     while True:
16         value = random.randint(20,30)
17         print(value)
18
19         node = client.get_node(nodeId)
20         value = float(value)
21         node.set_data_value(value)
22         time.sleep(10)
```

OPC UA Client Write Example



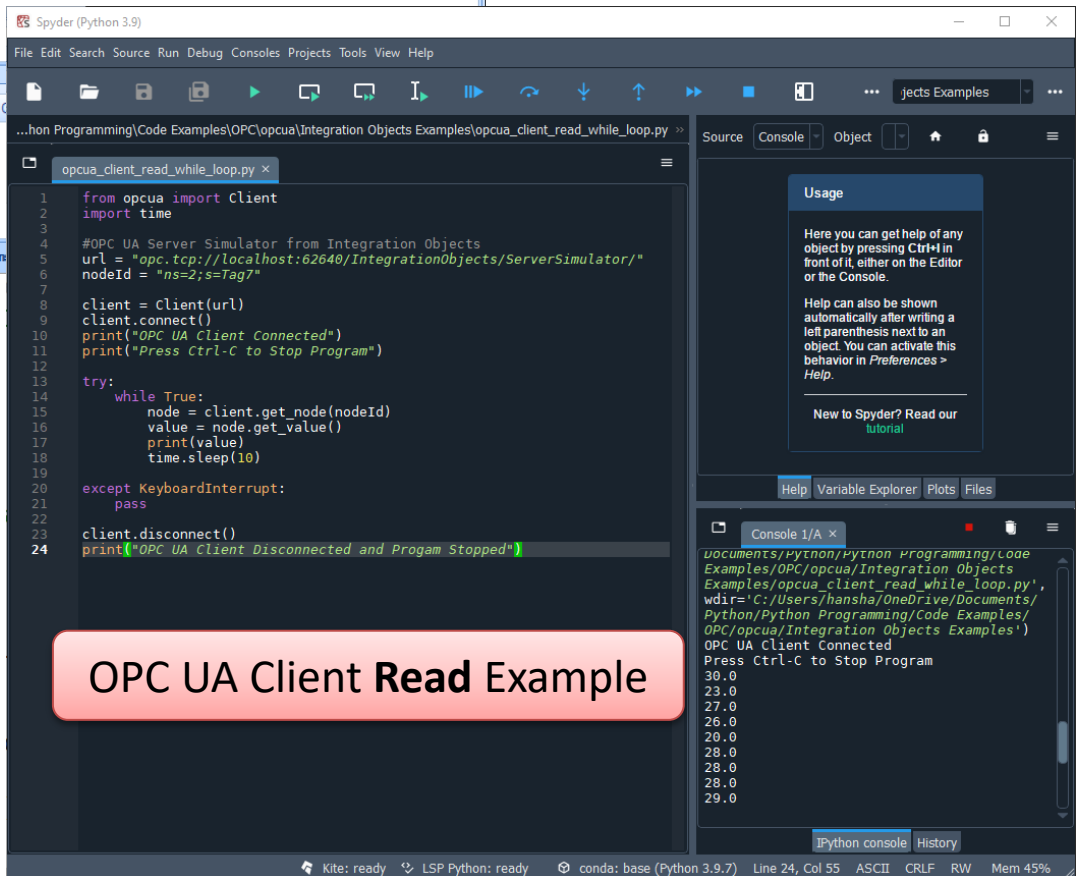
OPC UA Server Simulator

File Settings Help

Server Endpoints URLs: opc.tcp://xps13hph:62640/IntegrationObjects/ServerSimulator

SessionId	Name	User	Last Contact
Pure Python Client...	Anonymous	ns=3;i=896433817	10:40:05
Pure Python Client...	Anonymous	ns=3;i=896433873	10:40:03

# Spyder Python Editor



```
1 from opcua import Client
2 import time
3
4 #OPC UA Server Simulator from Integration Objects
5 url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
6 nodeId = "ns=2;s=Tag7"
7
8 client = Client(url)
9 client.connect()
10 print("OPC UA Client Connected")
11 print("Press Ctrl-C to Stop Program")
12
13 try:
14     while True:
15         node = client.get_node(nodeId)
16         value = node.get_value()
17         print(value)
18         time.sleep(10)
19
20 except KeyboardInterrupt:
21     pass
22
23 client.disconnect()
24 print("OPC UA Client Disconnected and Program Stopped")
```

OPC UA Client Read Example

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

New to Spyder? Read our [tutorial](#)

Console 1/A x

```
documents/python/python_programming/code_examples/opcua/integration_objects_examples/opcua_client_read_while_loop.py', wdirc='C:/Users/hansha/OneDrive/Documents/python/python_programming/code_examples/opcua/integration_objects_examples')
OPC UA Client Connected
Press Ctrl-C to Stop Program
30.0
23.0
27.0
26.0
20.0
28.0
28.0
28.0
29.0
```

## OPC UA Client **Write** Example

```
from opcua import Client
import time
import random

#OPC UA Server Simulator from Integration Objects
url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
nodeId = "ns=2;s=Tag7"

client = Client(url)
client.connect()
print("OPC UA Client Connected")
print("Press Ctrl-C to Stop Program")

try:
    while True:
        value = random.randint(20,30)
        print(value)

        node = client.get_node(nodeId)
        value = float(value)
        node.set_data_value(value)
        time.sleep(10)

except KeyboardInterrupt:
    pass

client.disconnect()
print("OPC UA Client Disconnected and Program Stopped")
```

## OPC UA Client Read Example

```
from opcua import Client
import time

#OPC UA Server Simulator from Integration Objects
url = "opc.tcp://localhost:62640/IntegrationObjects/ServerSimulator/"
nodeId = "ns=2;s=Tag7"

client = Client(url)
client.connect()
print("OPC UA Client Connected")
print("Press Ctrl-C to Stop Program")

try:
    while True:
        node = client.get_node(nodeId)
        value = node.get_value()
        print(value)
        time.sleep(10)

except KeyboardInterrupt:
    pass

client.disconnect()
print("OPC UA Client Disconnected and Program Stopped")
```



# OPC UA Python Client GUI

# Installation

- OPC UA Python Client GUI
- <https://github.com/FreeOpcUa/opcua-client-gui>

# Installation using Thonny

The screenshot shows the Thonny package manager interface. At the top, the title bar reads "Manage packages for C:\Users\hansha\AppData\Local\Programs\Thonny\python.exe". A search bar contains the text "opcu-client" and a "Search on PyPI" button. On the left, a list of packages is shown, with "opcu-client" selected and highlighted in blue. The main area displays the details for "opcu-client":

- Installed version:** 0.8.4
- Installed to:** <C:\Users\hansha\AppData\Roaming\Python\Python310\site-packages>
- Latest stable version:** 0.8.4
- Summary:** OPC-UA Client GUI
- Author:** Olivier R-D
- Homepage:** <https://github.com/FreeOpcUa/opcu-client-gui>
- PyPI page:** <https://pypi.org/project/opcu-client/>
- Requires:** PyQt5, asyncua, opcu-widgets (>=0.6.0)

At the bottom, there are buttons for "Upgrade", "Uninstall", "...", and "Close".

This depends on which Python version you have installed on your pc





Manage packages for C:\Users\hansha\AppData\Local\Programs\Thonny\python.exe



pyqtgraph

Search on PyPI

<INSTALL>

adafruit-board-toolkit  
aiofiles  
aiosqlite  
astroid  
asttokens  
asyncua  
bcrypt  
bitstring  
cffi  
colorama  
contourpy  
cryptography  
cyclur  
deprecation  
dill  
dnspython  
docutils  
ecdsa  
esptool  
fonttools  
guizero  
isort

## pyqtgraph

**Latest stable version:** 0.13.3

**Summary:** Scientific Graphics and GUI Library for Python

**Author:** Luke Campagnola

**Homepage:** <http://www.pyqtgraph.org>

**PyPI page:** <https://pypi.org/project/pyqtgraph/>

**Requires:** numpy (>=1.20.0)

Install

...

Close

You may also need to install “pyqtgraph”

# Location of opcua-client.exe

This depends on which Python version you have installed on your pc

The screenshot shows a Windows File Explorer window with the address bar set to `AppData > Roaming > Python > Python310 > Scripts`. The file `opcua-client.exe` is highlighted in blue. A red box highlights the path in the address bar, and another red box highlights the file name in the list.

Name	Date modified	Type	Size
f2py.exe	2023-02-15 15:01	Application	106 KB
fonttools.exe	2023-02-15 15:01	Application	106 KB
opcua-client.exe	2023-11-29 11:30	Application	106 KB
pyftmerge.exe	2023-02-15 15:01	Application	106 KB
pyftsubset.exe	2023-02-15 15:01	Application	106 KB
pylupdate5.exe	2023-11-29 11:30	Application	106 KB
pyrcc5.exe	2023-11-29 11:30	Application	106 KB
pyuic5.exe	2023-11-29 11:30	Application	106 KB
ttx.exe	2023-02-15 15:01	Application	106 KB
uabrowse.exe	2023-11-29 11:30	Application	106 KB
uacall.exe	2023-11-29 11:30	Application	106 KB
uaclient.exe	2023-11-29 11:30	Application	106 KB
uadiscover.exe	2023-11-29 11:30	Application	106 KB
uageneratestructs.exe	2023-11-29 11:30	Application	106 KB
uahistoryread.exe	2023-11-29 11:30	Application	106 KB
uals.exe	2023-11-29 11:30	Application	106 KB
uaread.exe	2023-11-29 11:30	Application	106 KB
uaserver.exe	2023-11-29 11:30	Application	106 KB
uasubscribe.exe	2023-11-29 11:30	Application	106 KB
uawrite.exe	2023-11-29 11:30	Application	106 KB

# OPC UA Python Client GUI

The screenshot displays the FreeOpcUa Client GUI. The main window title is "FreeOpcUa Client" and the address bar shows "opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator". The interface includes a tree view on the left, an attributes table on the right, a graph area, and a log window at the bottom.

**Tree View:**

DisplayName	BrowseName	NodeId
> Tag11	0:Tag11	ns=2;s=1:Tag11
> Tag12	0:Tag12	ns=2;s=1:Tag12
> Tag13	0:Tag13	ns=2;s=1:Tag13
> Tag14	0:Tag14	ns=2;s=1:Tag14
> Tag15	0:Tag15	ns=2;s=1:Tag15
> Tag16	0:Tag16	ns=2;s=1:Tag16
> Tag17	0:Tag17	ns=2;s=1:Tag17
> Tag18	0:Tag18	ns=2;s=1:Tag18
> Tag19	0:Tag19	ns=2;s=1:Tag19
> Tag2	0:Tag2	ns=2;s=1:Tag2
> Tag20	0:Tag20	ns=2;s=1:Tag20
> Tag3	0:Tag3	ns=2;s=1:Tag3
> Tag4	0:Tag4	ns=2;s=1:Tag4
> Tag5	0:Tag5	ns=2;s=1:Tag5
> Tag6	0:Tag6	ns=2;s=1:Tag6
> Tag7	0:Tag7	ns=2;s=1:Tag7
> Tag8	0:Tag8	ns=2;s=1:Tag8
> Tag9	0:Tag9	ns=2;s=1:Tag9
> Methods	2:MethodCalls	ns=2;s=MethodCalls
> Real Time Data	2:Realtimedata	ns=2;s=Realtimedata
> Types	0:Types	i=86
> Views	0:Views	i=87

**Attributes Table:**

Attribute	Value	Data Type
AccessLevel	CurrentRead, ...	Byte
ArrayDimension	None	Null
BrowseName	0:Tag7	QualifiedName
Data Type	Double	NodeId
Description	None	Null
DisplayName	LocalizedText(LocalizedText	LocalizedText
Historizing	True	Boolean
MinimumSampling	-1.0	Double
NodeClass	2	Int32

**Graph:**

Number of Points: 30, Intervall [s]: 5

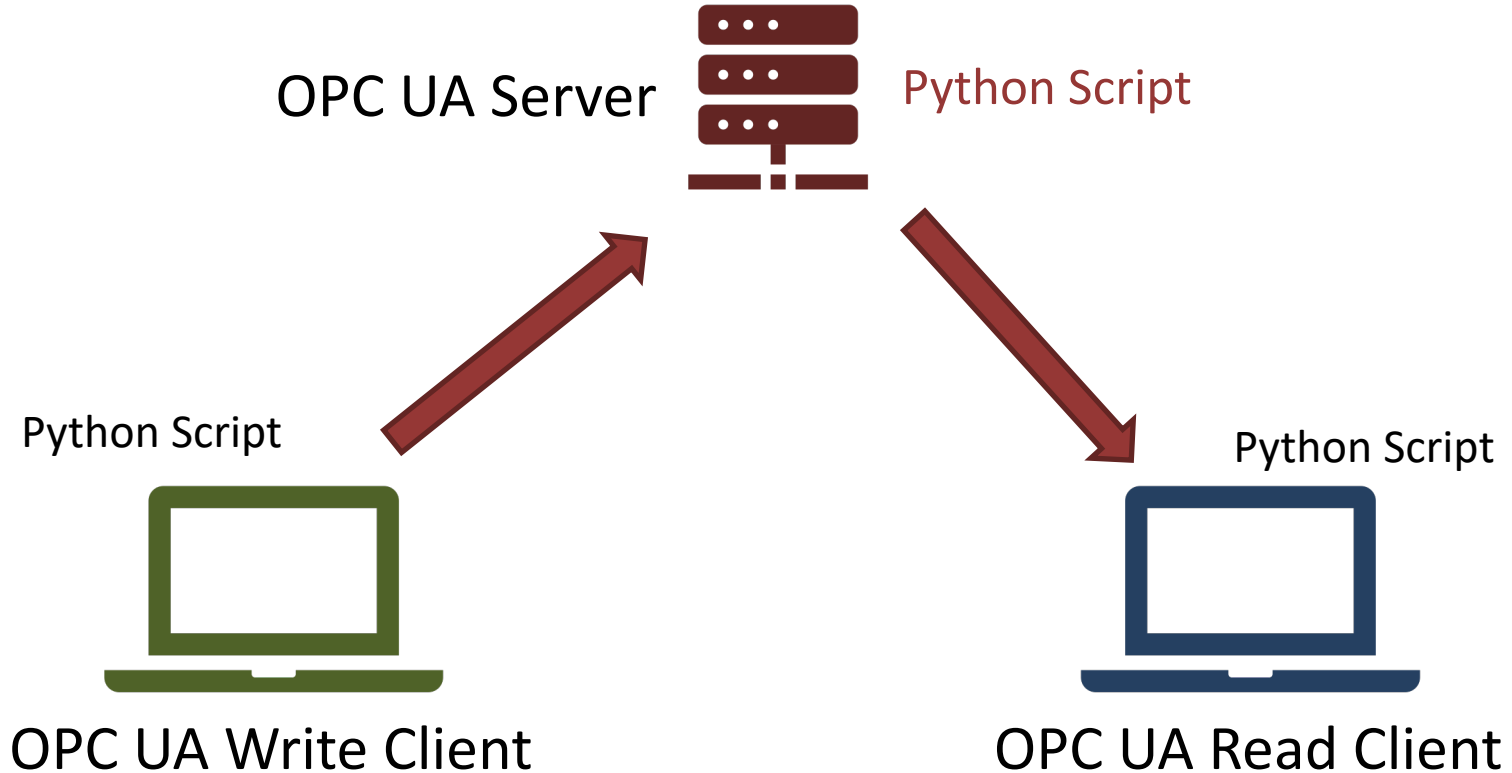
**Log Window:**

```
uaclient.uaclient - INFO - Connecting to opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator with parameters None, None, None, None')
asynqua.client.client - WARNING - Deprecated since spec 1.04, call load_data_type_definitions')
asynqua.client.client - WARNING - Deprecated since spec 1.04, call load_data_type_definitions')
```



# Python OPC UA Server Examples

# Python OPC UA Server



# OPC UA Server

```
from opcua import Server

server = Server()
server_url = "opc.tcp://127.0.0.1:1234"
server.set_endpoint(server_url)

name = "opcuapython"
namespace = server.register_namespace(name)

node = server.get_objects_node()
param = node.add_object(namespace, "Sensors")

var = param.add_variable(namespace, "Temperature", 0)
var.set_writable()

server.start()
```

# OPC UA Server

```
from opcua import Server
import time
from datetime import datetime

server = Server()
server_url = "opc.tcp://127.0.0.1:1234"
server.set_endpoint(server_url)

name = "opcuapython"
namespace = server.register_namespace(name)
node = server.get_objects_node()
param = node.add_object(namespace, "Sensors")
var = param.add_variable(namespace, "Temperature", 0)
var.set_writable()

server.start()
print("OPC UA Server Started")
print("Press Ctrl-C to Stop Program")

try:
    while True:
        now = datetime.now()
        current_time = now.strftime("%H:%M:%S")
        print("OPC UA Server Running", current_time)
        value = var.get_value()
        print("Current Value:", value)
        time.sleep(1)

except KeyboardInterrupt:
    pass

server.stop()
print("OPC UA Server Stopped")
```

# Test Connection to Server

FreeOpcUa Client

Actions Settings

opc.tcp://127.0.0.1:1234

DisplayName	BrowseName	NodId
Root	0:Root	i=84
Objects	0:Objects	i=85
Server	0:Server	i=2253
Sensors	2:Sensors	ns=2;i=1
Temperature	2:Temperature	ns=2;i=2
Types	0:Types	i=...
Views	0:Views	i=...

## OPC UA Client from Integration Objects

The screenshot shows the 'Integration Objects' OPC UA Client interface. The 'Attribute Value' table is highlighted, showing the following data:

Attribute	Value
NodeId	ns=2;i=2
NodeClass	Variable
BrowseName	2:Temperature
DisplayName	Temperature
Description	Temperature
WriteMask	0
UserWriteMask	0
Value	0
DataType	Int64
ValueRank	Scalar
ArrayDimensions	System.UInt32[]
AccessLevel	Readable   Writeable
UserAccessLevel	Readable   Writeable
MinimumSamplingInterval	Continuous
Historizing	False

The message log at the bottom shows the following messages:

Message Type	Timestamp	Message
[Control]	2023-11-29 13:57:08	Write operation of the variable [ns=2;i=2] succeeded.
[Control]	2023-11-29 13:56:56	Read operation of the variable [ns=2;i=2] succeeded.
[Control]	2023-11-29 13:56:45	A session "Session0" with the Endpoint [opc.tcp://127.0.0.1:1234/- [None:None:Binary]] was successfully created.
[Error]	2023-11-29 13:56:38	Session creation failed. Exception: Invalid URI: Invalid port specified.

OPC UA Python Client GUI

NodId



# Get Nodeld

```
from opcua import Client

#OPC UA Server Python
url = "opc.tcp://127.0.0.1:1234"

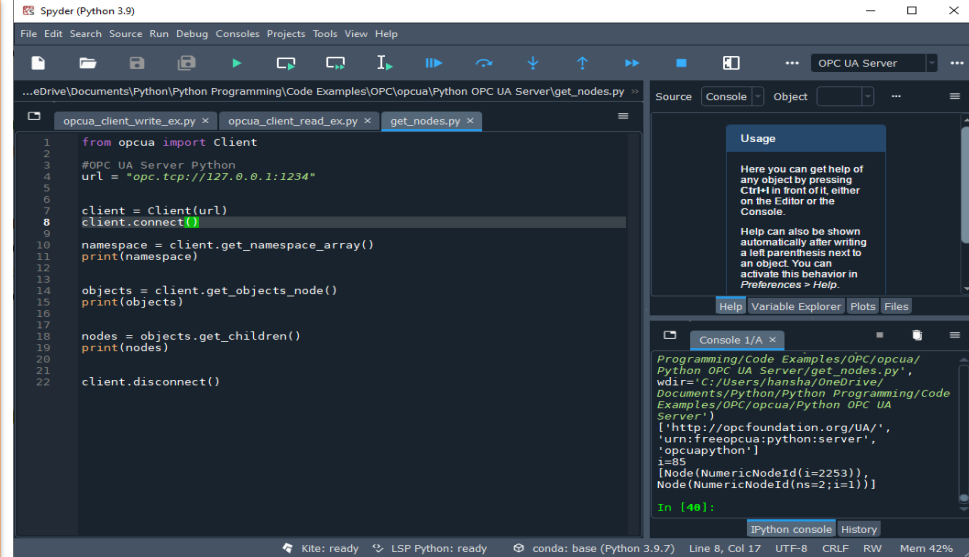
client = Client(url)
client.connect()

namespace = client.get_namespace_array()
print(namespace)

objects = client.get_objects_node()
print(objects)

nodes = objects.get_children()
print(nodes)

client.disconnect()
```



The screenshot shows the Spyder Python IDE with a script named 'get\_nodes.py' open. The script performs the following actions:

- Imports the Client class from the opcua module.
- Defines the OPC UA server URL as 'opc.tcp://127.0.0.1:1234'.
- Creates a Client object and connects to the server.
- Retrieves the namespace array and prints it.
- Retrieves the objects node and prints it.
- Retrieves the children of the objects node and prints them.
- Disconnects the client.

The console output shows the following NodeId:

```
[Node(NumericNodeId(i=2253)),  
Node(NumericNodeId(ns=2;i=1))]  
In [40]:
```

This gives:  
NodeId = "ns=2;i=2"

# OPC UA Client Write Example

```
from opcua import Client

#OPC UA Server Python
url = "opc.tcp://127.0.0.1:1234"
nodeId = "ns=2;i=2"

client = Client(url)
client.connect()

node = client.get_node(nodeId)
value = float(20)
node.set_data_value(value)

client.disconnect()
```

# Write - Alternative Solution

```
from opcua import Client

#OPC UA Server Python
url = "opc.tcp://127.0.0.1:1234"

client = Client(url)
client.connect()

root = client.get_root_node()
nodeId = root.get_child(["0:Objects", "2:Sensors", "2:Temperature"])
print("NodeId:", nodeId)

node = client.get_node(nodeId)
value = float(20)
node.set_data_value(value)

client.disconnect()
```

# OPC UA Client Read Example

```
from opcua import Client

#OPC UA Server Python
url = "opc.tcp://127.0.0.1:1234"
nodeId = "ns=2;i=2"

client = Client(url)
client.connect()

node = client.get_node(nodeId)
value = node.get_value()
print(value)

client.disconnect()
```

# Read - Alternative Solution

```
from opcua import Client

#OPC UA Server Python
url = "opc.tcp://127.0.0.1:1234"

client = Client(url)
client.connect()

root = client.get_root_node()
nodeId = root.get_child(["0:Objects", "2:Sensors", "2:Temperature"])
print("NodeId:", nodeId)

node = client.get_node(nodeId)
value = node.get_value()
print("Value[°C]:", value)

client.disconnect()
```

```

1 from opcua import Server
2 import time
3 from datetime import datetime
4
5 server = Server()
6 server_url = "opc.tcp://127.0.0.1:1234"
7 server.set_endpoint(server_url)
8
9 name = "opcuapython"
10 namespace = server.register_namespace(name)
11 node = server.get_objects_node()
12 param = node.add_object(namespace, "Sensors")
13 var = param.add_variable(namespace, "Temperature", 0)
14 var.set_writable()
15
16 server.start()
17 print("OPC UA Server Started")
18 print("Press Ctrl-C to Stop Program")

```

```

Shell x
Current Value: 20.0
OPC UA Server Running 13:39:06
Current Value: 20.0
OPC UA Server Running 13:39:07
Current Value: 20.0
OPC UA Server Running 13:39:08
Current Value: 20.0
OPC UA Server Running 13:39:09
Current Value: 20.0

```

## OPC UA Server

```

1 from opcua import Client
2
3 #OPC UA Server Python
4 url = "opc.tcp://127.0.0.1:1234"
5 nodeId = "ns=2;i=2"
6
7 client = Client(url)
8 client.connect()
9
10 node = client.get_node(nodeId)
11 value = float(20)
12 node.set_data_value(value)
13
14 client.disconnect()

```

### Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in [Preferences > Help](#).

New to Spyder? Read our [tutorial](#)

```

Programming\Code Examples\OPC\opcua\
Python OPC UA Server\
opcua_client_read_ex.py', wdir='C:/
Users/hansha/OneDrive/Documents/Python/
Python Programming/Code Examples/OPC/
opcua/Python OPC UA Server')
20.0

```

```

In [20]: runfile('C:/Users/hansha/
OneDrive/Documents/Python/Python
Programming/Code Examples/OPC/opcua/
Python OPC UA Server/
opcua_client_write_ex.py', wdir='C:/
Users/hansha/OneDrive/Documents/Python/
Python Programming/Code Examples/OPC/
opcua/Python OPC UA Server')

```

```

In [29]:

```

## OPC UA Clients

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

