

Tips, Tricks, and Tidbits

- [Known limitations](#)
- [Moving beyond a single flat file](#)
- [How to fix pyntree installing as UNKNOWN-0.0.0](#)

Known limitations

- Attempting to retrieve a name which is also a Node property will not work as intended (ex: `_values`) when doing so explicitly (`Node._values`), but using the `get` method will yield the desired result. For example, if you have a data point/key with a sub-attribute of `_values`, running `Node._values` will work as [described here](#) rather than returning that sub-attribute. However, running `Node.get('_values')` will work as intended. In code, this looks like:

```
x = Node()
x._values = 1
x._values
Output: ['values']
x.get('_values')
Output: <pyntree.Node object at memory_location>
```

Moving beyond a single flat file

For more complex projects, a flat-file database may not be the way to go. Here is one example of how you could store data in a more distributed way:

- db folder
 - users
 - user1.pyn
 - user2.pyn
 - etc
 - transactions
 - 001.pyn
 - 002.pyn
 - etc
 - config.json

Python implementation:

```
from pyntree import Node
config = Node('db/config.json')

# On request with argument "transaction_id"
data = Node(f'db/transactions/{transaction_id}')
return data()
```

How to fix pyntree installing as UNKNOWN-0.0.0

```
pip install --upgrade pip wheel setuptools
```