PythonNET GrasshopperTemplate

Documentation

Credit: This documentation is largely copied from
https://github.com/JonasFeron/PythonConnectedGrasshopperTemplate

Developer: Jonas FERON

Opensource: https://github.com/JonasFeron/PythonNETGrasshopperTemplate

PythonNET GrasshopperTemplate

Overview

Introduction

Why using PythonNETGrasshopperTemplate ?

•**Seamless Grasshopper Plugin Development** – Combine C# and Python 3 (latest version) to create multiple custom Grasshopper components that leverage the strengths of both languages.

•**Optimized Execution and Data Exchange** – Run Python scripts within Grasshopper components, efficiently transfer data, and use Python libraries like NumPy and Pandas without delays.

•**A Practical Alternative to Traditional Grasshopper Scripting** – Provides a structured and scalable approach to integrating Python into Grasshopper beyond basic scripting.

•**Ideal for Complex and Scalable Grasshopper Plug-ins** – Easily manage and develop multiple interconnected components within Visual Studio, improving organization and maintainability.

•**Conclusion**: PythonNETGrasshopperTemplate simplifies Grasshopper plugin development, making it easier to integrate and manage Python scripts within C# workflows.

C# (in Visual Studio) for custom Grasshopper Plug-In

But what about Python ?

Manage multiple grasshopper components written in C# in Visual Studio:

- Follow the official tutorial : Grasshopper - Your First Component
- Use the official Visual Studio Grasshopper Template
- Tip: In Grasshopper Developer Settings: do not forget to add path to your plug in

→ What if you need specific Python Librairies (like NumPy for scientific computing) ?

→ What if you already have your custom python scripts,
and want to use Grasshopper as a user interface ?

Python for custom Grasshopper component(s)

But not in Visual Studio and hence not for complex plug-in

- Follow the official tutorial Grasshopper Scripting: Python to develop your custom Grasshopper components in Python
- Python scripts are written directly within Grasshopper, not Visual Studio

→ Difficulties to manage a full plug-in made of multiple components

→ How to develop Grasshopper plug-in in Visual Studio for Python ? (no solution)

→ PythonNETGrasshopperTemplate

PythonNETGrasshopperTemplate

is based on

[Python.NET](#) ✚ [GrasshopperTemplate](#)

Visual Studio

Grasshopper - test

Custom Plug-in

custom components

C# Code in Visual Studio for this custom component

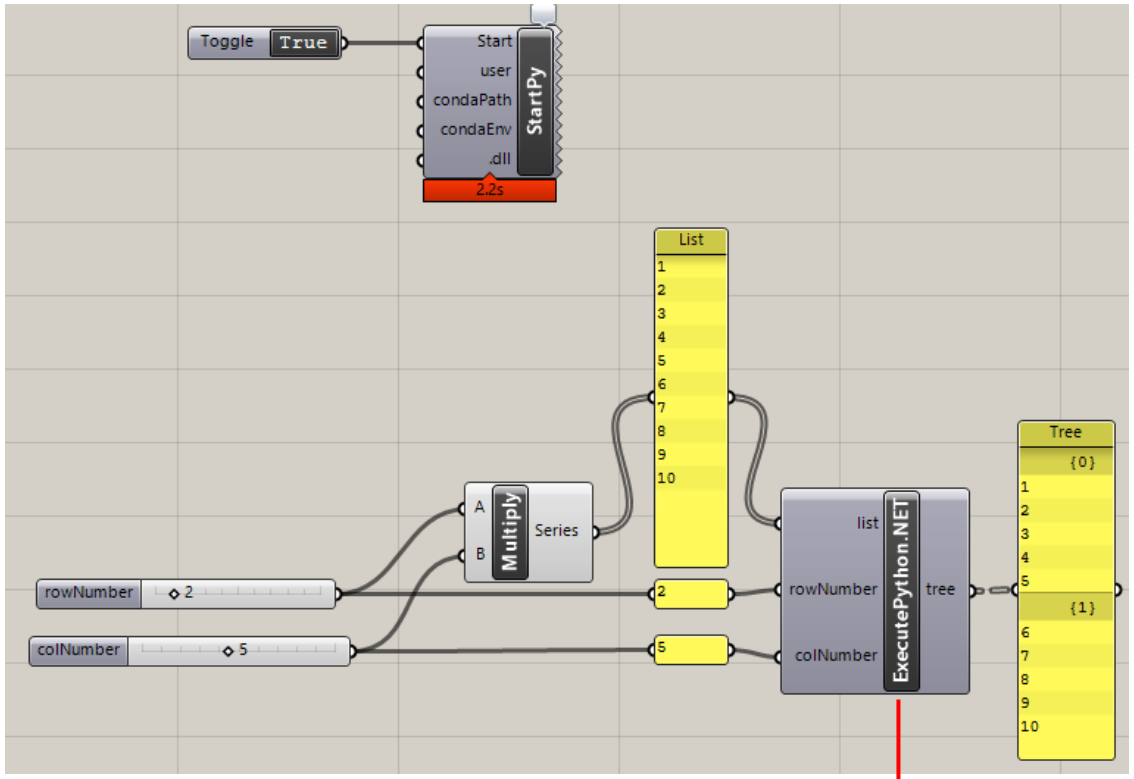
call Python.NET

```
# complex_script.py
import numpy as np

def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

Step 1) Initialize Python.NET

Step 2) Execute Python scripts in no time from Grasshopper




```python
# complex_script.py
import numpy as np


def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

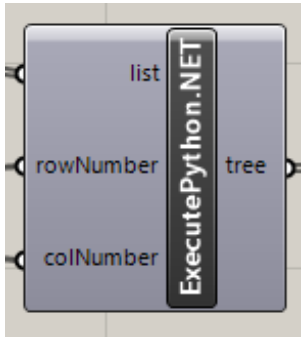
PythonNETGrasshopperTemplate

Manages multiple C# and Python Components in Visual Studio

Visual Studio

a Grasshopper component written in C#

call Python.NET and import « complex_script.py »




```python
# complex_script.py
import numpy as np

def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

PythonNET GrasshopperTemplate

Conclusion

Python.NET Vs PythonConnect GrasshopperTemplate

[Grasshopper Template]

OR

https://github.com/JonasFeron/PythonNETGrasshopperTemplate

https://github.com/JonasFeron/PythonConnectedGrasshopperTemplate

- **Both Python.NET and PythonConnect require data conversion** between C# and Python.
- Regarding data transfer between C# and Python, **Python.NET is faster than PythonConnect** because:
  - Python.NET transfers data through shared memory
  - PythonConnect transfers data through read/write dataFiles.txt

PythonNET GrasshopperTemplate

Appendices

Getting started with Python.NET

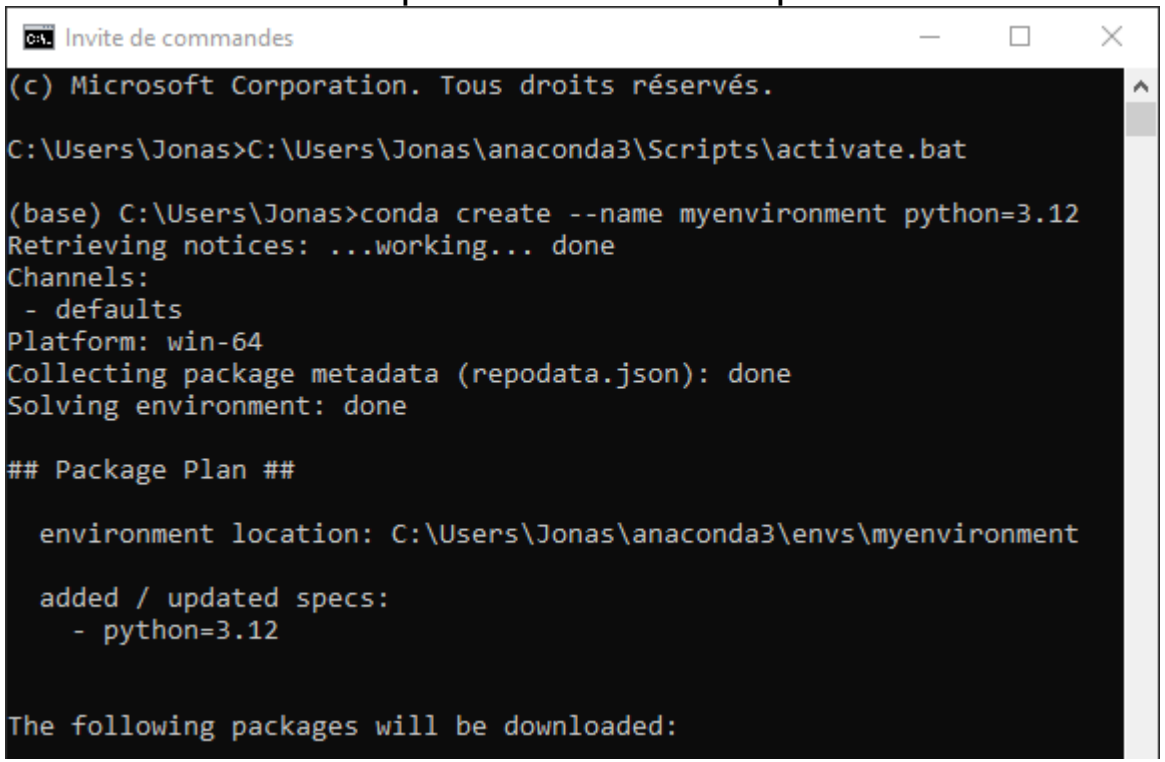
1) Download and Install Anaconda
2) Follow Python.NET Wiki

Getting started with Python.NET

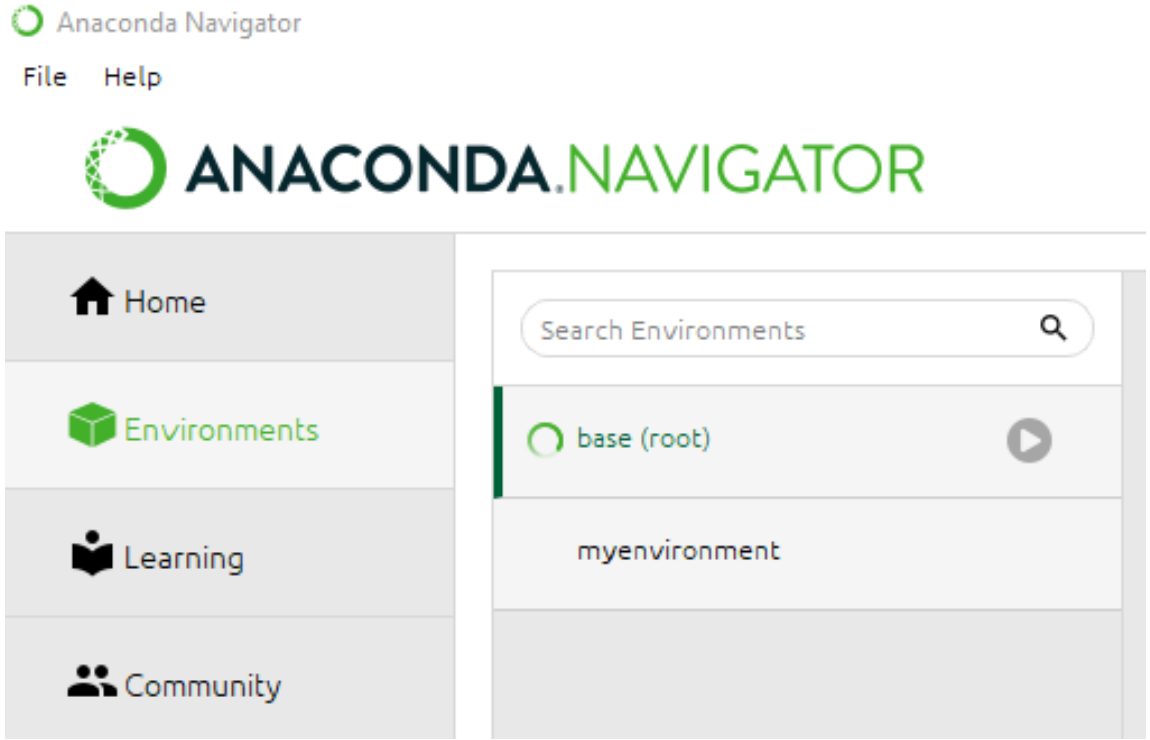
2) Manage python virtual environment

1. Use (base) conda environment
2. Or create a new environment for specific python version

via Command Prompt / Anaconda Prompt



Or via Anaconda Navigator App

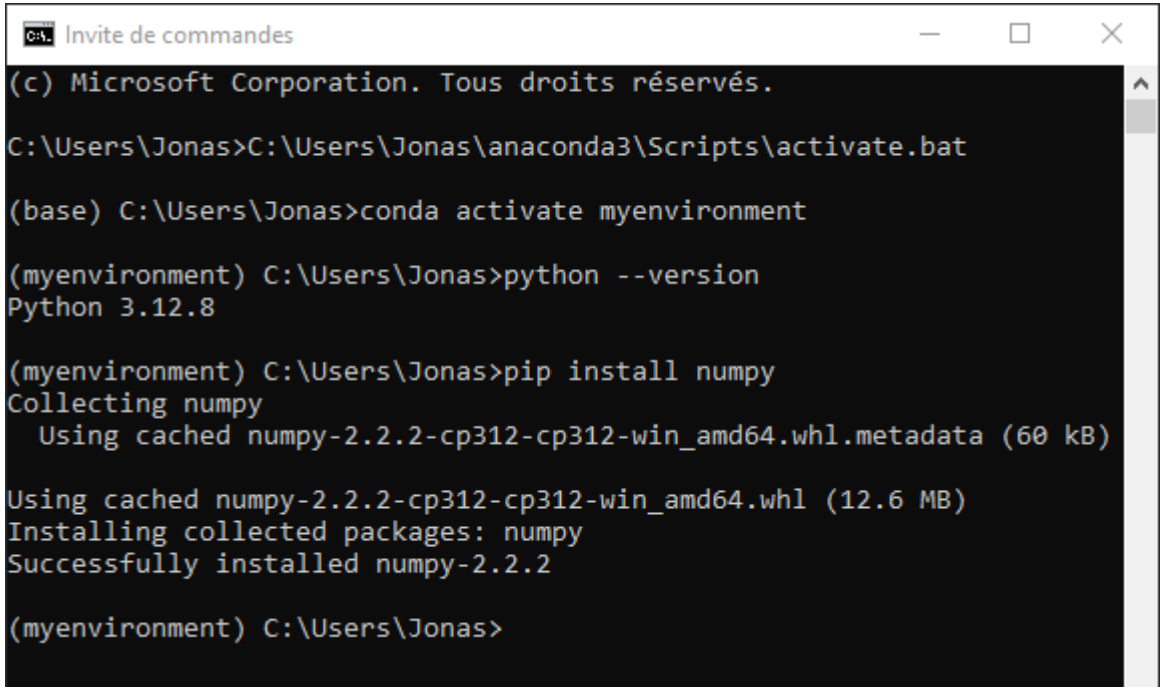


Getting started with Python.NET

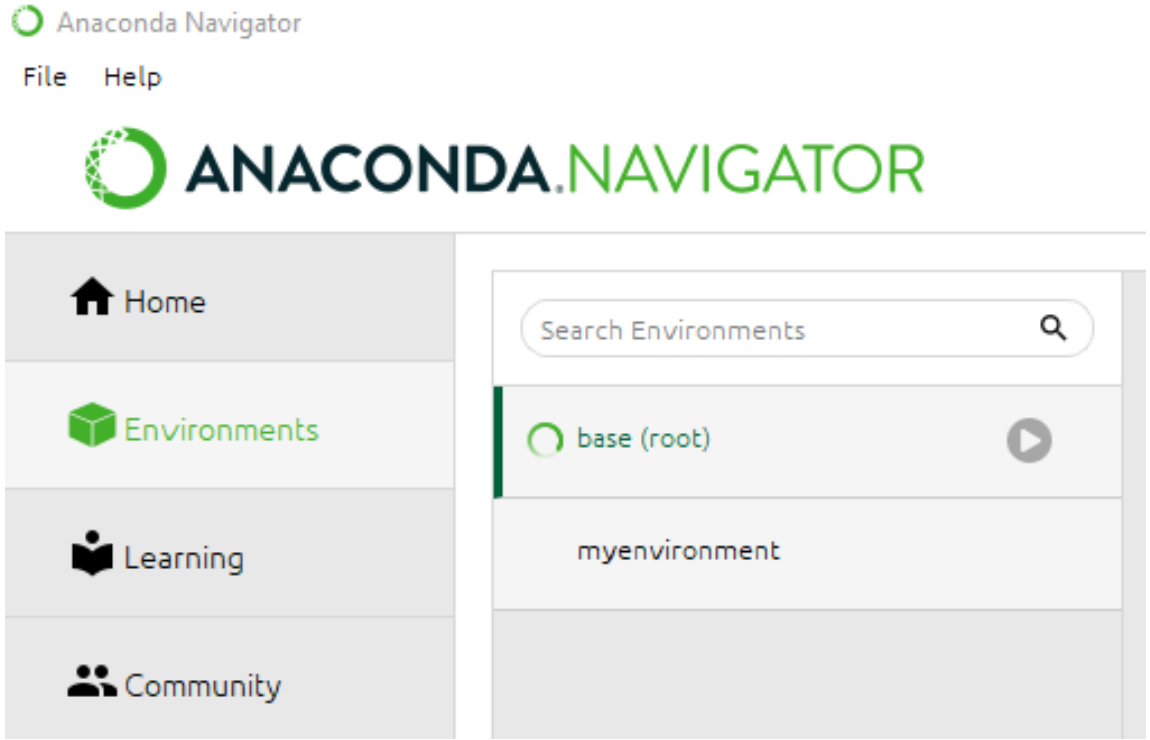
3) Install required python librairies in the environment

Including https://pypi.org/project/pythonnet/

via Command Prompt / Anaconda Prompt



Or via Anaconda Navigator App



Getting started with PythonNETGrasshopperT.

4) From JonasFeron/PythonNETGrasshopperTemplate

- Clone the main branch of the Github repository, locally on your computer (GitHub Desktop helps)
- Open file src/PythonNETGrasshopperTemplate.sln using Visual Studio
- Run the project, which launches Rhino in debug Mode
- Follow the official tutorial : Grasshopper - Your First Component