



Center for Geographic Analysis
Harvard University



Open for Innovation
KNIME

Build Python Geospatial tools In KNIME

Lingbo Liu

Center for Geographic Analysis, Harvard University

lingboliu@fas.Harvard.edu

Download KNIME Analytics Platform 4.7 Nightly Build

Download the latest nightly build:

<https://www.knime.com/nightly-build-downloads>

KNIME Analytics Platform

See the full list of changes in the [changelog](#).

Windows	
KNIME Analytics Platform Nightly build for Windows (installer)	64bit
KNIME Analytics Platform Nightly build for Windows (self extracting archive)	64bit
KNIME Analytics Platform Nightly build for Windows (zip archive)	64bit (SHA-256)
Linux	
KNIME Analytics Platform Nightly build for Linux	64bit (SHA-256)
Mac OS X	
KNIME Analytics Platform Nightly build for Mac OS X (dmg)	64bit

C > Software (D:) > Software > knime_470 >			
Name	Date modified	Type	Size
bundling	7/13/2022 9:25 PM	File folder	
configuration	7/13/2022 9:19 PM	File folder	
dropins	7/13/2022 7:57 AM	File folder	
features	7/13/2022 9:26 PM	File folder	
knimespace	7/13/2022 10:49 PM	File folder	
licenses	7/13/2022 7:58 AM	File folder	
p2	7/13/2022 7:57 AM	File folder	
plugins	7/13/2022 9:26 PM	File folder	
artifacts.xml	7/13/2022 9:26 PM	XML Document	165 KB
knime.exe	7/12/2022 5:50 PM	Application	414 KB
knime.ini	7/13/2022 10:51 PM	Configuration sett...	2 KB
knimec.exe	7/12/2022 5:52 PM	Application	129 KB
knime-workspace.zip	7/12/2022 5:50 PM	Compressed (zipp...	7,520 KB
LICENSE.TXT	7/12/2022 5:50 PM	Text Document	1 KB
README.txt	7/12/2022 5:50 PM	Text Document	2 KB

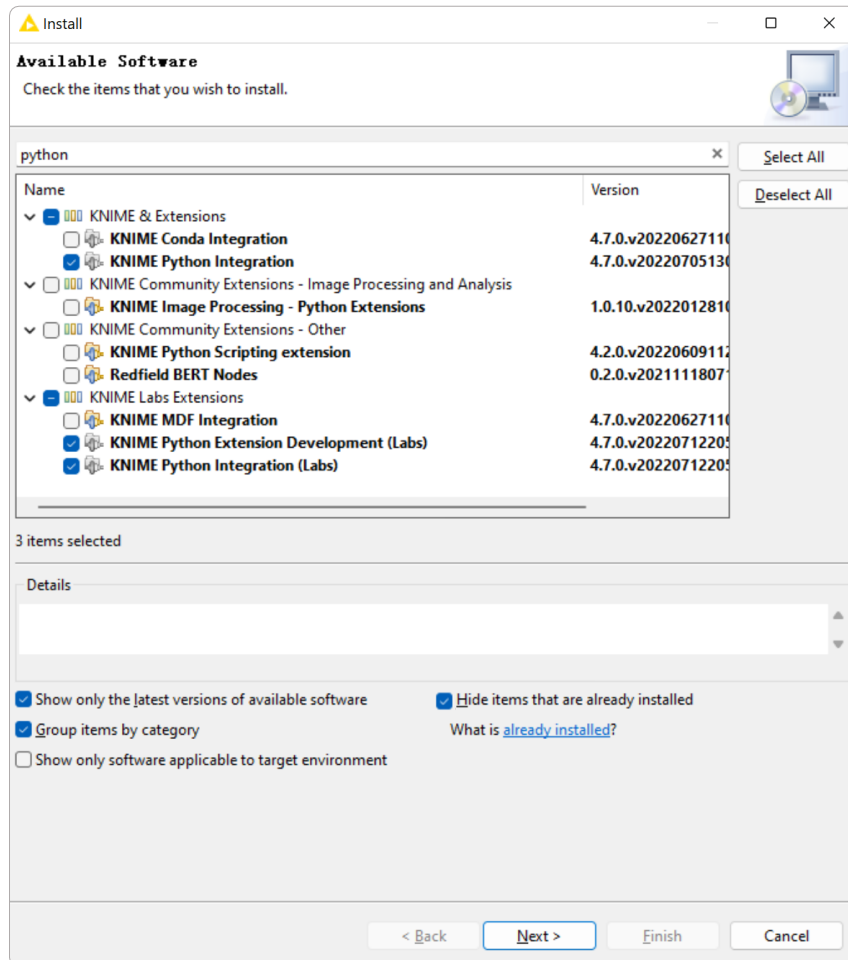
Install Related KNIME Python Extensions

KNIME -> File-> Install KNIME Extensions

Search for Python and select

KNIME Python Extension Development (Labs)

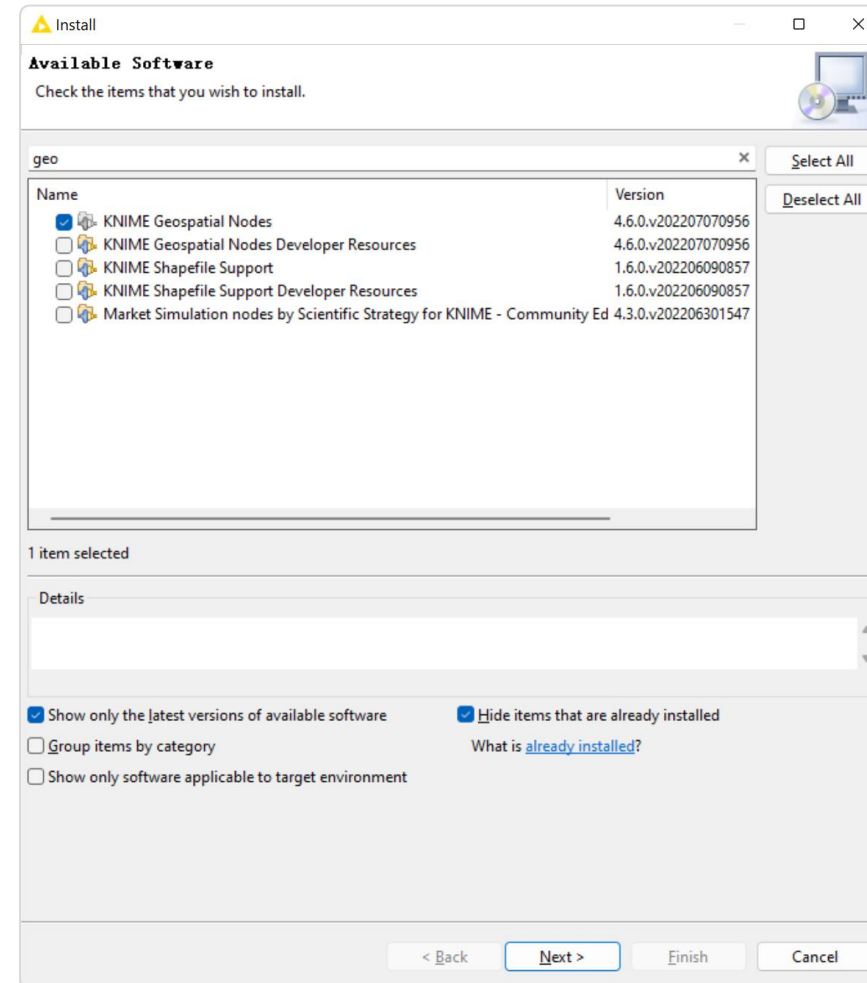
KNIME Python Integration (Labs)



Disable the “Group items by category” option

Search for geo and select

KNIME Geospatial Nodes



Anaconda Prompt—Build New Python Environment for Geospatial Nodes

Anaconda Prompt: Setting up new conda environment *my_python_env* also include the *geopandas* package e.g.

conda create -n my_python_env python=3.9 knime-python-base knime-extension geopandas -c knime -c conda-forge

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\Lingbo Liu>conda create -n my_python_env python=3.9 knime-python-base knime-extension geopandas -c knime -c conda-forge
```

Anaconda Prompt: activate *my_python_env*

conda info

Record the env location path : D:\ProgramData\Anaconda3\envs\my_python_env

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\Lingbo Liu>activate my_python_env
(my_python_env) C:\Users\Lingbo Liu>conda info

active environment : my_python_env
active env location : D:\ProgramData\Anaconda3\envs\my_python_env
shell level : 2
user config file : C:\Users\Lingbo Liu\.condarc
populated config files : C:\Users\Lingbo Liu\.condarc
conda version : 4.13.0
conda-build version : 3.21.6
python version : 3.9.7.final.0
```

Configure KNIME Python Setting

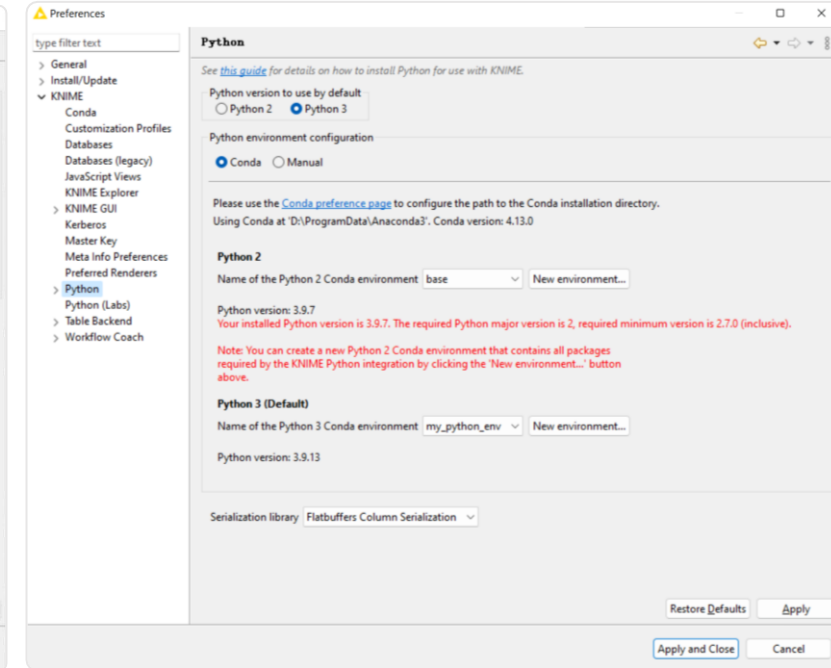
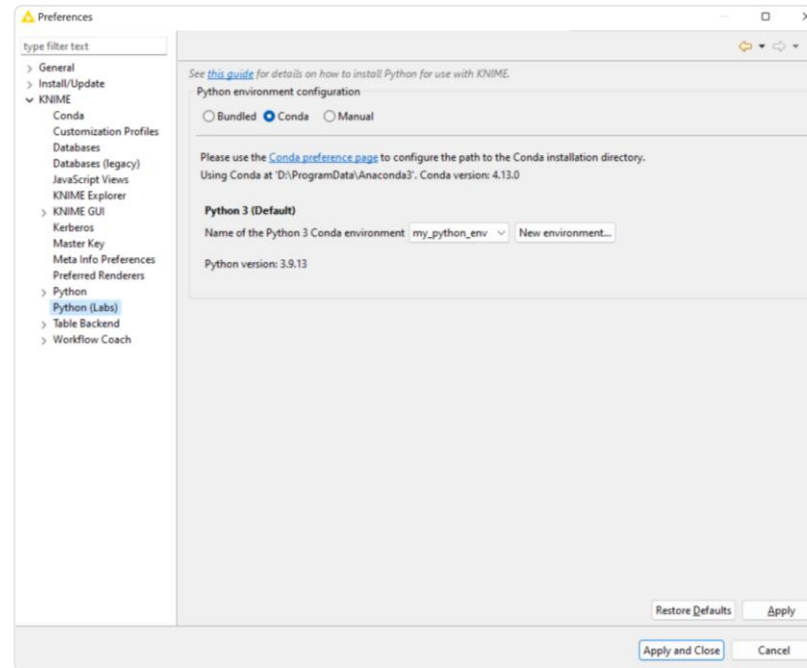
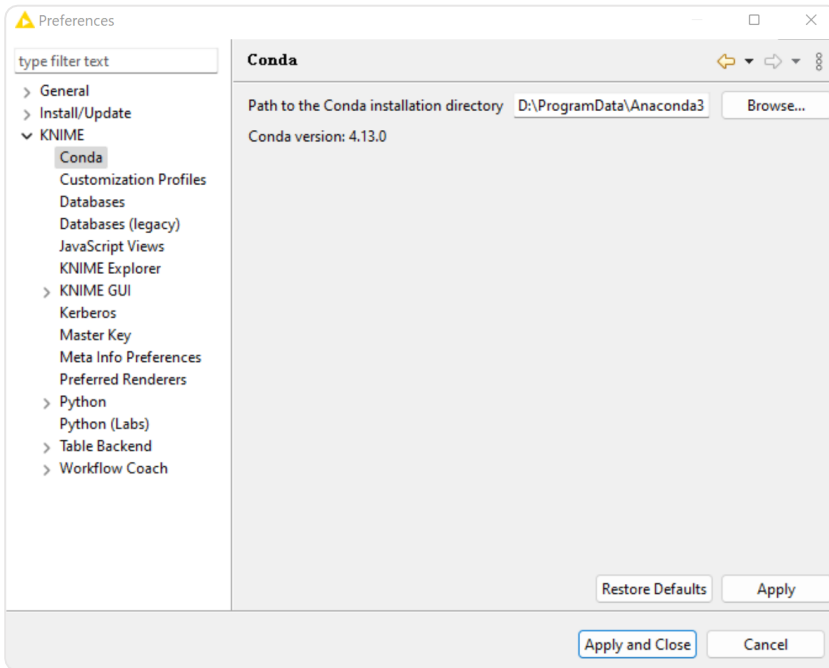
Configure KNIME Python setting

KNIME -> File-> Preference> KNIME> Conda Choose anaconda directory

KNIME -> File-> Preference> KNIME> Python(Labs) Choose *my_python_env*

***KNIME -> File-> Preference> KNIME> Python(Labs)** Choose *my_python_env*

** This step is not necessary unless you installed KNIME Python Integration*



Extract geo.zip and set config.yml

Extract geo.zip, it contains a subfolder geo_extension

Name	Date modified	Type	Size
geo_extension	7/13/2022 10:15 PM	File folder	
config.yml	7/13/2022 9:41 PM	Yaml Source File	1 KB
example_code.py	7/13/2022 9:38 PM	Python Source File	4 KB
my_conda_env.yml	7/13/2022 9:38 PM	Yaml Source File	1 KB
README.md	7/13/2022 9:38 PM	Markdown Source...	1 KB

```
D: > Software > knime_470 > knimespace > geo > ! my_conda_env.yml
1 name: my_python_env
2 channels:
3   - knime
4   - conda-forge
5 dependencies:
6   - packaging
7   - python=3.9
8   - knime-extension
9   - knime-python-base
10  - geopandas
```

Change the path in src and conda_env_path

```
! config.yml X
D: > Software > knime_470 > knimespace > geo > ! config.yml
1 org.knime.geo: # {group_id}.{name} from the knime.yml
2 src: D:\Software\knime_470\knimespace\geo\geo_extension # Path to folder containing the extension files
3 conda_env_path: D:\ProgramData\Anaconda3\envs\my_python_env # Path to the Python environment to use
4 debug_mode: true # Optional line, if set to true, it will always use the latest changes of execute/configure, when that method is used within the KNIME Analytics Platform
```

Python tools in the subfolder geo_extension

Name	Date modified	Type	Size
__pycache__	7/13/2022 9:47 PM	File folder	
geo_category.py	7/13/2022 9:38 PM	Python Source File	1 KB
icon.png	7/13/2022 9:38 PM	PNG File	1 KB
knime.yml	7/13/2022 9:38 PM	Yaml Source File	1 KB
knime_geo.py	7/13/2022 9:38 PM	Python Source File	1 KB
knime_geo_length.py	7/13/2022 10:22 PM	Python Source File	2 KB
LICENSE.TXT	7/13/2022 9:38 PM	Text Document	37 KB
transformer_nodes.py	7/13/2022 9:38 PM	Python Source File	2 KB

Configure knime.ini and All set

Add the following line to your *knime.ini* file which should point to *config.yml* file within the extracted folder

`-Dknime.python.extension.config=D:\Software\knime_470\knimespace\geo\config.yml`

The image shows three windows from a Windows desktop environment:

- File Explorer:** Displays the contents of the `C:\Software (D:) > Software > knime_470` directory. The file list includes folders like `bundling`, `configuration`, `dropins`, `features`, `knimespace`, `licenses`, `p2`, `plugins`, and files like `artifacts.xml`, `knime.exe`, `knime.ini`, `knimec.exe`, `knime-workspace.zip`, `LICENSE.TXT`, and `README.txt`.
- Notepad:** The title bar reads "knime.ini - Notepad". The menu bar shows "File", "Edit", and "View". The text area contains the following configuration lines:

```
--add-opens=java.base/java.net=ALL-UNNAMED
--add-opens=java.base/java.nio=ALL-UNNAMED
--add-opens=java.base/java.nio.channels=ALL-UNNAMED
--add-opens=java.base/java.util=ALL-UNNAMED
--add-opens=java.base/sun.nio.ch=ALL-UNNAMED
--add-opens=java.base/sun.nio=ALL-UNNAMED
--add-opens=java.desktop/javafx.swing.plaf.basic=ALL-UNNAMED
--add-opens=java.base/sun.net.www.protocol.http=ALL-UNNAMED
-Xmx8g
-Dorg.eclipse.swt.browser.IEVersion=11001
-Dsun.awt.noerasebackground=true
-Dequinox.statechange.timeout=30000
-Darrow.enable_unsafe_memory_access=true
-Darrow.memory.debug allocator=false
-Darrow.enable_null_check_for_get=false
--add-opens=java.security.jgss/sun.security.jgss.krb5=ALL-UNNAMED
--add-exports=java.security.jgss/sun.security.jgss=ALL-UNNAMED
--add-exports=java.security.jgss/sun.security.jgss.spi=ALL-UNNAMED
--add-exports=java.security.jgss/sun.security.krb5.internal=ALL-UNNAMED
--add-exports=java.security.jgss/sun.security.krb5=ALL-UNNAMED
-Dknime.python.extension.config=D:\Software\knime_470\knimespace\geo\config.yml
```

The status bar at the bottom indicates "Ln 1, Col 1", "100%", and "Windows (CRLF)".
- Node Repository:** A window titled "Node Repository" showing a tree view of KNIME nodes. The tree is expanded to show the "Geospatial" category, which includes "Compute Length", "CRS Transformer", and "Compute Area".

Add New nodes to Geospatial

Create New Python file in the `geo_extension` folder, and read the nodes by import in `knime_geo.py`

> Software (D:) > Software > knime_470 > knimespace > geo > geo_extension >		
Name	Date modified	Type
__pycache__	7/14/2022 8:34 AM	File folder
geo_category.py	7/13/2022 9:38 PM	Python Source File
icon.png	7/13/2022 9:38 PM	PNG File
knime.yml	7/13/2022 9:38 PM	Yaml Source File
knime_geo.py	7/14/2022 8:34 AM	Python Source File
knime_geo_length.py	7/13/2022 10:22 PM	Python Source File
LICENSE.TXT	7/13/2022 9:38 PM	Text Document
transformer_nodes.py	7/13/2022 9:38 PM	Python Source File

```
D: > Software > knime_470 > knimespace > geo > geo_extension > knime_geo.py > ...
1  import logging
2  import knime_extension as knext
3  import pandas as pd
4  import geopandas as gp
5  import geo_category
6  import transformer_nodes
7  import knime_geo_length
8
9  LOGGER = logging.getLogger(__name__)
10
11
12  @knext.node(name="Compute Area", node_type=knext.NodeType.MANIPULATOR, icon_path="icon.png", category=geo_category.category)
13  @knext.input_table(name="Geo table", description="Table with geometry column to compute the area for")
14  @knext.output_table(name="Geo table with area", description="Geo input table with additional area column")
15  class ComputeAreaNode:
16      """
17      This node computes the area of a geo cell.
18      """
19
20      def configure(self, configure_context, input_schema_1):
21          return input_schema_1.append(knext.Column(knext.double(), "area"))
22
23      def execute(self, exec_context, input_1):
24          gdf=gp.GeoDataFrame(input_1.to_pandas())
25          gdf['area']=gdf.area
26          #why do we need to convert the GeoDataFrame into a pandas data frame???
27          return knext.Table.from_pandas(pd.DataFrame(gdf))
28
```

Special Thanks to Tobias Koetter and Carsten Haubold



Center for Geographic Analysis
Harvard University



Open for Innovation
KNIME

Build Python Geospatial tools In KNIME

Special Thanks to Tobias Koetter and Carsten Haubold