SIDRA BATCH Python sample program using SIDRA API

Harry Cai

Presentation at the SIDRA API Workshop
Melbourne, November 2018

Updated: 2 May 2019

SIDRA SOLUTIONS

PRESENTATION OBJECTIVES / CONTENTS

SIDRA API Python Sample Program

- Project files
- Python source code

SIDRA SOLUTIONS

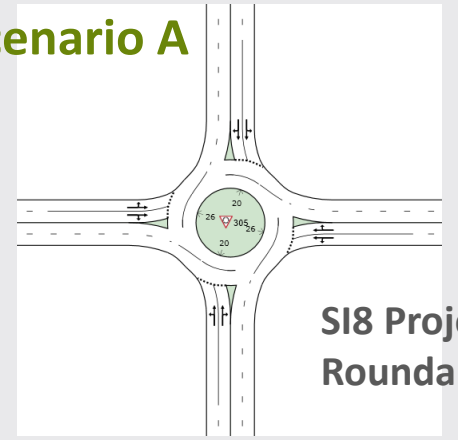
Project Files in This Sample

- **Four Proposed Designs** for one Intersection
  - Roundabout - **Scenario A**
  - Signalised - **Scenario A**
  - Roundabout - **Scenario B**
  - Signalised - **Scenario B**
- Each Project file has two Sites
  - AM Peak Volumes
  - PM Peak Volumes
- Each Site has been set with a corresponding **Category**

Project Files in This Sample

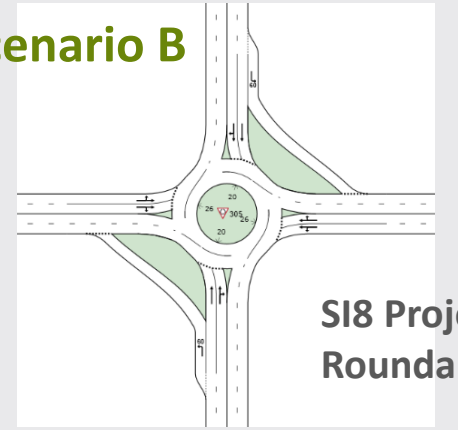
Roundabout

Scenario A



SI8 Project file:
Roundabout - Scenario A.sip8

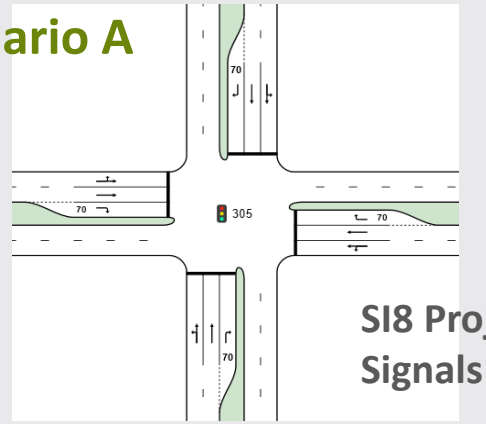
Scenario B



SI8 Project file:
Roundabout - Scenario B.sip8

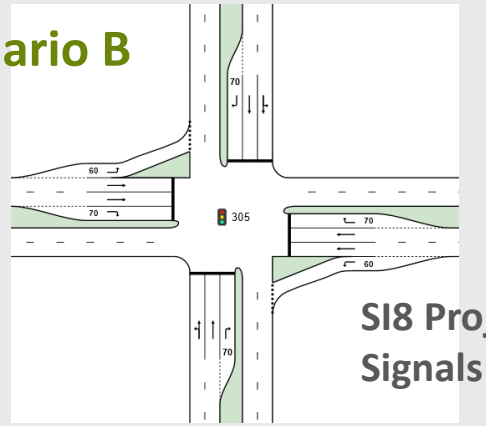
Signalised

Scenario A



SI8 Project file:
Signals - Scenario A.sip8

Scenario B



SI8 Project file:
Signals - Scenario B.sip8

SIDRA SOLUTIONS

Python Source Code

Prerequisites

- Install Python for Windows Extensions Package “pypiwin32”
  - Install through PIP: run the command “pip install pypiwin32”, or
  - Download the package from https://pypi.python.org/pypi/pypiwin32
  - Note: There is a similar package called “pywin32”. Do not use this package.

Python Source Code

Prerequisites

- Prepare SIDRA API-Python Package “SIAPIComPy.py”
  - Download Python Sample program from SIDRA Support website -> Resources home -> SIDRA INTERSECTION -> SIDRA UTILITIES and API. The package is included in this sample program. Or
  - Use the Python tool “makepy.py” to generate it.
    1. Open "Command Prompt“
    2. Run "<Python_install_folder>\Lib\site-packages\win32com\client\makepy.py“
    3. Select "SIDRA INTERSECTION 8.0 API (8.0)" in the popup "Select Library" window, click "OK“
    4. Copy the generated "A073B4B7-AFE5-4EE8-A7D3-6FFEB4FFEB09x0x8x0.py" file to the folder of your Python SIDRA API Program file and rename it to "SIAPIComPy.py".

Python Source Code

Process Sites from Multiple Project Files

- Import required packages

```
import win32com.client as w32c
import pythoncom
import SIAPIComPy
```

- Create SIAPI object, wrap it with SIAPIComPy.ISIAPI class

```
self.siapi = SIAPIComPy.ISIAPI(w32c.Dispatch("SIDRASolutions.SI.API.SIAPI"))
```

The wrapping is important. Without it, Python does not know what attributes can be achieved from SIAPI object. We need to use wrapping on every object that comes from SIDRA API.

- Open a Project file and get the Project object

```
self.siapi.OpenProject(fileName)
project = SIAPIComPy.ISIAPIProject(self.siapi.Project)
   # wrap it with ISIAPIProject
```

Python Source Code

Process Sites from Multiple Project Files

- Get a collection of Sites from the Project

```
sites = SIAPIComPy.ISIAPISites(project.Sites) # wrap it with ISIAPISites
```

- Go through each Site by index

```
for siteIndex in range(sites.Count):
  site = SIAPIComPy.ISIAPISite(sites.Item_2(siteIndex))
    # site index starts from 0
```

- Process the Site and get the Site Output data

```
  site.Process()
  if site.Outputset:
    outputset = SIAPIComPy.ISIAPIOutputset(site.Outputset)
```

Python Source Code

Read Site Output Data

- Read Intersection Summary Output data for all Vehicle Movements on the Site

```
outputSiteVeh = SIAPIComPy.ISIAPIOutputSiteVehicle(outputset.OutputSiteVehicle)

myUIData.Flow_total = outputSiteVeh.Flow_total
myUIData.Capacity_effective = outputSiteVeh.Capacity_effective
myUIData.Deg_satn = outputSiteVeh.Deg_satn
# ...

self.UIProjectSummarySiteDataList.append(myUIData)
```

- The object "`myUIData`" is a local defined object to hold a copy of the Site's output data, and then is stored into a local List `UIProjectSummarySiteDataList`.
- When SIDRA API opens another Project, the current Project Output data will be removed. We make local copies of our interested data so that we can have the data from different Projects in the list.

Python Source Code

Generate Project Summary

- Go through a list of local copied data to generate a CSV file. The final output data in this example are grouped by the Category value.

Clean up

- In this example, we use Python code "`with … as …`" to create the example object `pySIAPI`, so that when the program finishes, its `__exit__()` method is called to clean up. Finally, call `pythoncom.CoUninitialize()` to release the COM resources.

```
with PySIAPI() as pySIAPI:
# ...
def __exit__(self, exc_type, exc_value, traceback):
  del self.siapi
# ...
finally:
  pythoncom.CoUninitialize()
```

END OF PRESENTATION

SIDRA SOLUTIONS