OpenDSS Training Workshop

Session III - Extending Capabilities of OpenDSS via a Programing Language

Paulo Radatz
EPRI Knoxville, TN

August 24, 2022

www.epri.com

Agenda

Session 1: Monday Aug. 22nd 11:30AM – 1:30PM EST

- > Distribution system basics, OpenDSS basics and scripting

Session 2: Tuesday Aug. 23rd 11:30AM – 1:30PM EST

- > Intro to OpenDSS-G, New Functionality in DSS, Advanced Topics

Session 3: Wednesday Aug. 24th 11:30AM – 1:30PM EST

- > Extending Capabilities of OpenDSS via a Programming Language

Session 4: Thursday Aug. 25th 11:30AM – 1:30PM EST

- > Applying DSS in R&D

Housekeeping

You'll get all resources (presentations and scripts) in the OpenDSS repository.

Link: https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Training/Virtual-2022/

You'll know when the video recordings are available via email.

We'll post it in the SourceForge discussion as well.

You'll earn Professional Development Hours (PDH).

You've have required it during the registration.

Any questions, please contact Arin Nichols (anichols@epri.com).

You can interact with us during the meeting by using the Webex Q&A feature.

We'll try to answer as many questions as possible.

Speaker

Paulo Radatz

Paulo Radatz has worked as an Engineer/Scientist at the Electric Power Research Institute (EPRI) in Knoxville, Tennessee, the USA, since 2019. He received his Master's and Bachelors's degrees in electrical engineering, emphasizing energy and automation, from the University of Sao Paulo, Sao Paulo, Brazil. He was awarded a prize for being the best bachelor's student at the Polytechnic School of the University of Sao Paulo (2015).

His work at EPRI focus on hosting capacity, mitigation options to increase hosting capacity, time-series hosting capacity, and DER modeling for QSTS simulation. Currently, he is the leading developer of the EPRI DRIVE tool and one of the OpenDSS developers.

He has seven years of experience with OpenDSS, having taught OpenDSS in several meetings, workshops, and training, including EPRI's OpenDSS virtual training. He is the creator of the world's largest OpenDSS YouTube channel: https://www.youtube.com/PauloRadatz. Currently, he lectures about OpenDSS in the Electric Power Distribution MBA course and the Power System Analysis with OpenDSS course, both at Polytechnic School of the University of Sao Paulo.

Agenda

- OpenDSS Flavors
- Why You Might Want to Write Some Code …
- Basics of Controlling OpenDSS
  - Load Circuit/Study
  - Manipulate Circuit, Elements, and Buses properties
  - Perform Simulations
  - Obtain Results
- Examples

OpenDSS Flavors

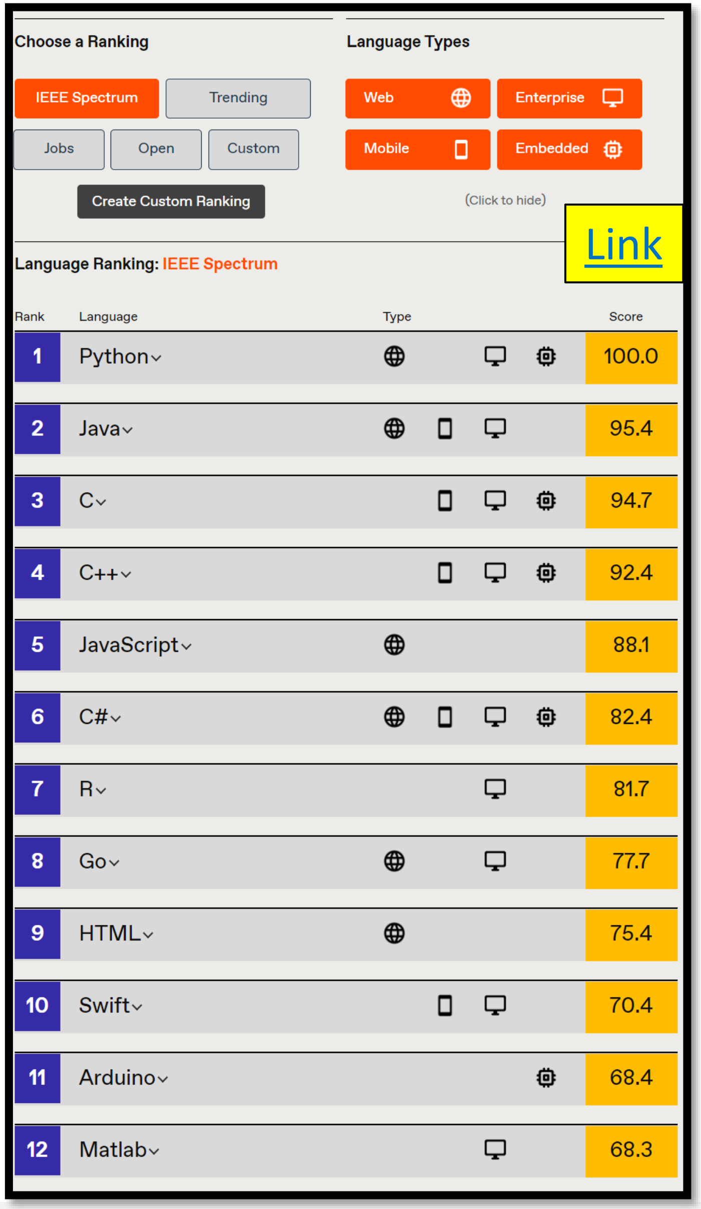
OpenDSS Versions

1.	OpenDSS.exe	Standalone EXE
2.	**OpenDSSEngine.dll**	***In-process* COM server**
3.	**OpenDSSDirect.dll**	**Direct Call DLL**
4.	OpenDSSCmd.exe	Standalone EXE command line

What Languages Can You use for Your Code?

- Excel VBA
- VB.net
- C#
- C/C++
- Delphi, Free Pascal
- MATLAB
- Python
- Java
- LabView
- R
- Fortran (for DLLs, with DirectDLL)
- Julia (with DirectDLL)

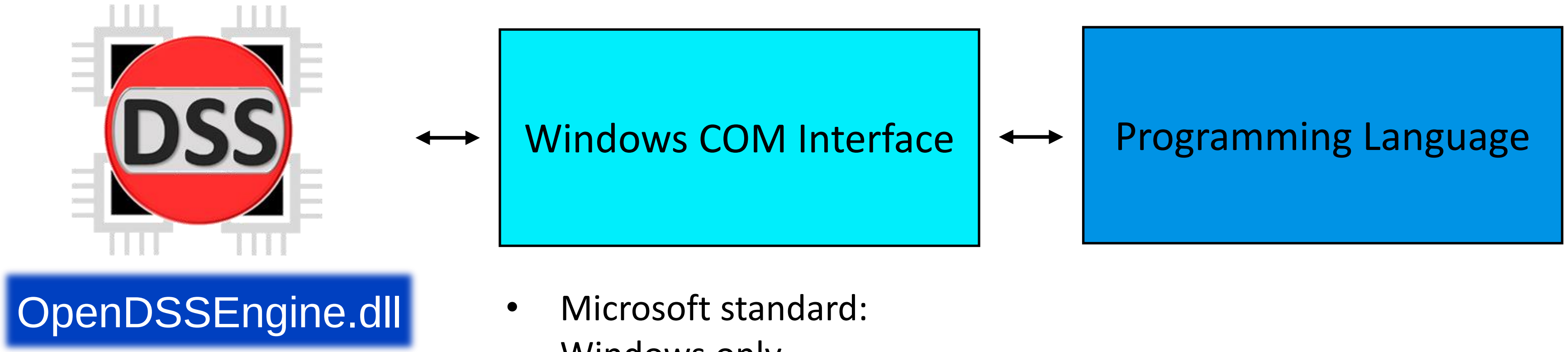
Opportunity to use your preferred programming language



Language Ranking: IEEE Spectrum

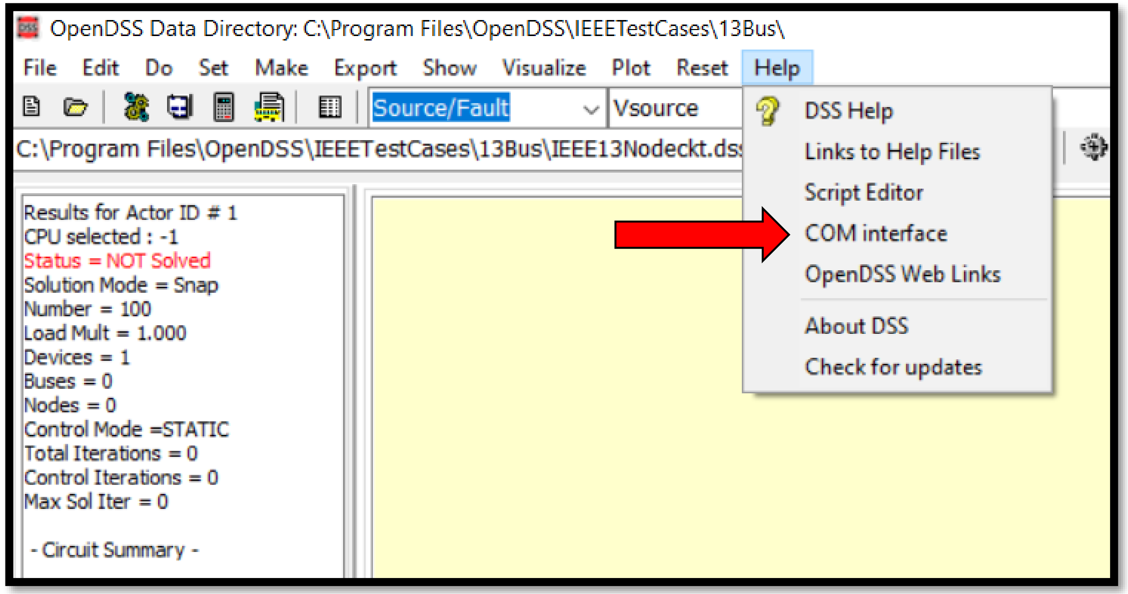
Rank	Language	Type	Score
1	Python	🌐 🖥 ⚙	100.0
2	Java	🌐 📱 🖥	95.4
3	C	📱 🖥 ⚙	94.7
4	C++	📱 🖥 ⚙	92.4
5	JavaScript	🌐	88.1
6	C#	🌐 📱 🖥 ⚙	82.4
7	R	🖥	81.7
8	Go	🌐 🖥	77.7
9	HTML	🌐	75.4
10	Swift	📱 🖥	70.4
11	Arduino	⚙	68.4
12	Matlab	🖥	68.3

Ways to Control OpenDSS – Option 1



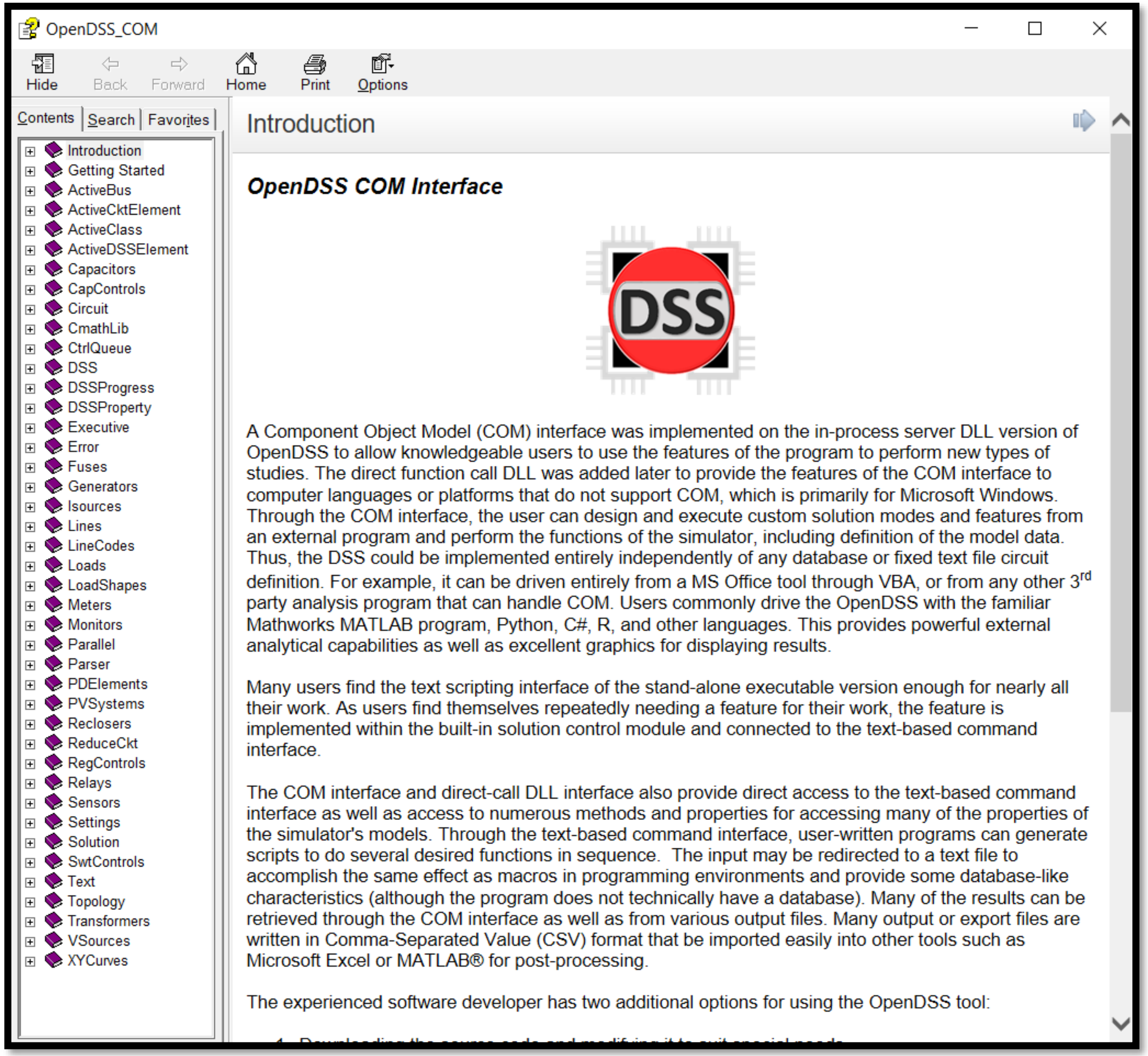
- Microsoft standard: Windows only.
- Well supported in MS Office as well as Python and other languages.

Ways to Control OpenDSS – Option 1



Results for Actor ID # 1
CPU selected : -1
Status = NOT Solved
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 1
Buses = 0
Nodes = 0
Control Mode =STATIC
Total Iterations = 0
Control Iterations = 0
Max Sol Iter = 0

- Circuit Summary -



Introduction

OpenDSS COM Interface

A Component Object Model (COM) interface was implemented on the in-process server DLL version of OpenDSS to allow knowledgeable users to use the features of the program to perform new types of studies. The direct function call DLL was added later to provide the features of the COM interface to computer languages or platforms that do not support COM, which is primarily for Microsoft Windows. Through the COM interface, the user can design and execute custom solution modes and features from an external program and perform the functions of the simulator, including definition of the model data. Thus, the DSS could be implemented entirely independently of any database or fixed text file circuit definition. For example, it can be driven entirely from a MS Office tool through VBA, or from any other 3rd party analysis program that can handle COM. Users commonly drive the OpenDSS with the familiar Mathworks MATLAB program, Python, C#, R, and other languages. This provides powerful external analytical capabilities as well as excellent graphics for displaying results.

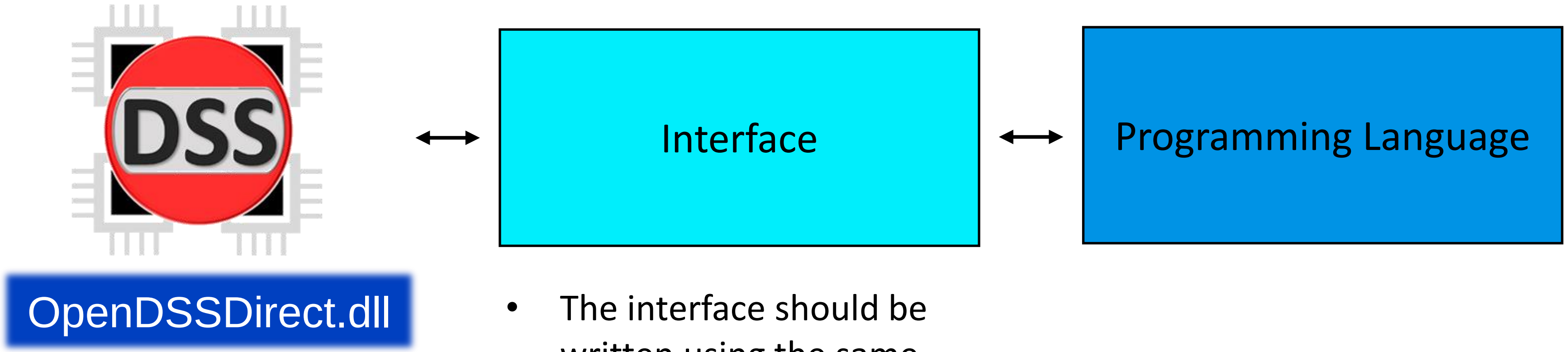
Many users find the text scripting interface of the stand-alone executable version enough for nearly all their work. As users find themselves repeatedly needing a feature for their work, the feature is implemented within the built-in solution control module and connected to the text-based command interface.

The COM interface and direct-call DLL interface also provide direct access to the text-based command interface as well as access to numerous methods and properties for accessing many of the properties of the simulator's models. Through the text-based command interface, user-written programs can generate scripts to do several desired functions in sequence. The input may be redirected to a text file to accomplish the same effect as macros in programming environments and provide some database-like characteristics (although the program does not technically have a database). Many of the results can be retrieved through the COM interface as well as from various output files. Many output or export files are written in Comma-Separated Value (CSV) format that be imported easily into other tools such as Microsoft Excel or MATLAB® for post-processing.

The experienced software developer has two additional options for using the OpenDSS tool:

1. Downloading the source code and modifying it to suit special needs.

Ways to Control OpenDSS – Option 2



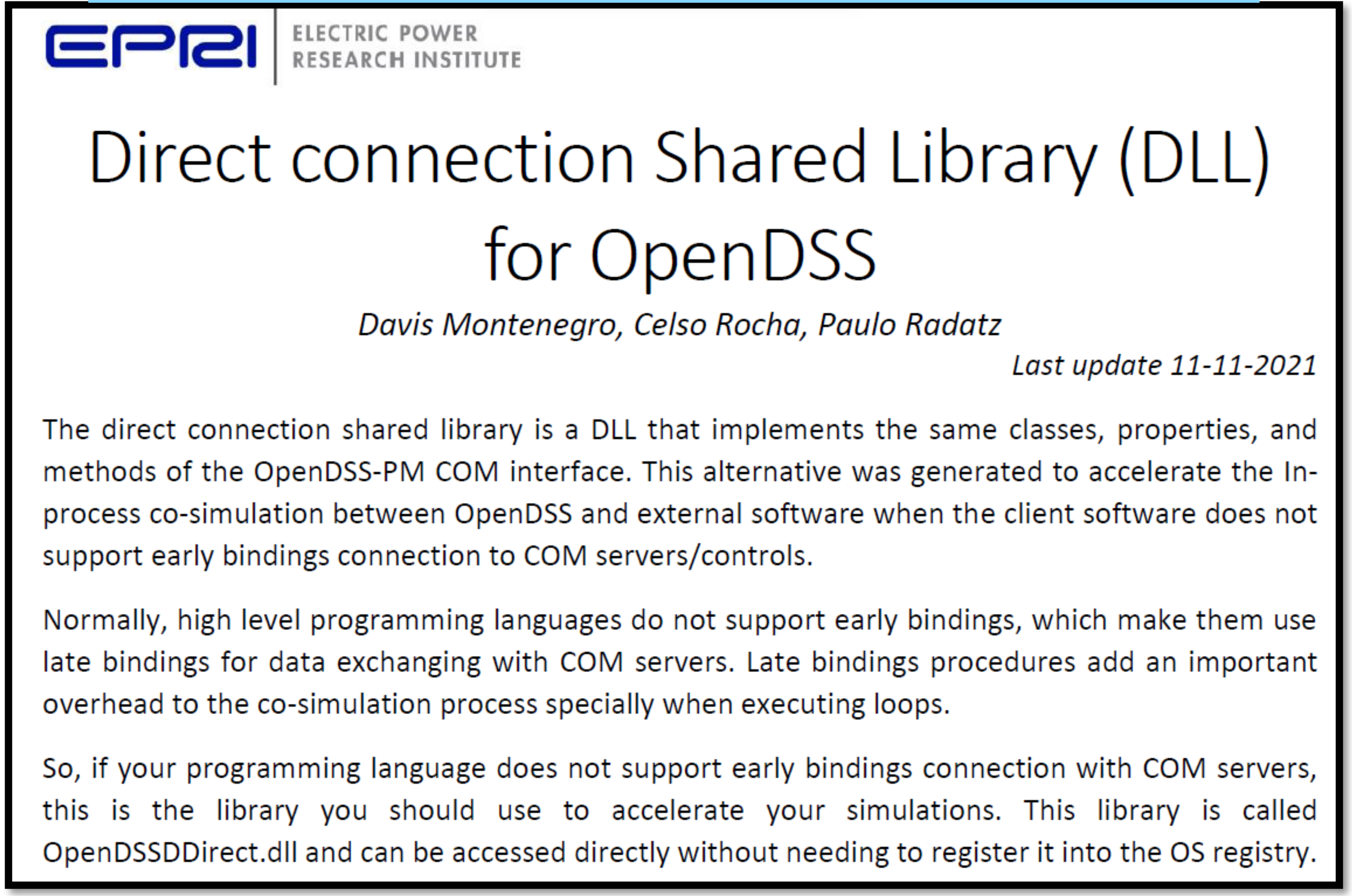
- The interface should be written using the same programming language that will control OpenDSS

- Example for Python:
  - py-dss-interface

Integrated Development Environment (IDE) Code Completion Feature

Ways to Control OpenDSS – Option 2

"C:\Program Files\OpenDSS\Doc\OpenDSS_Direct_DLL.pdf"



ELECTRIC POWER RESEARCH INSTITUTE

Direct connection Shared Library (DLL) for OpenDSS

Davis Montenegro, Celso Rocha, Paulo Radatz

Last update 11-11-2021

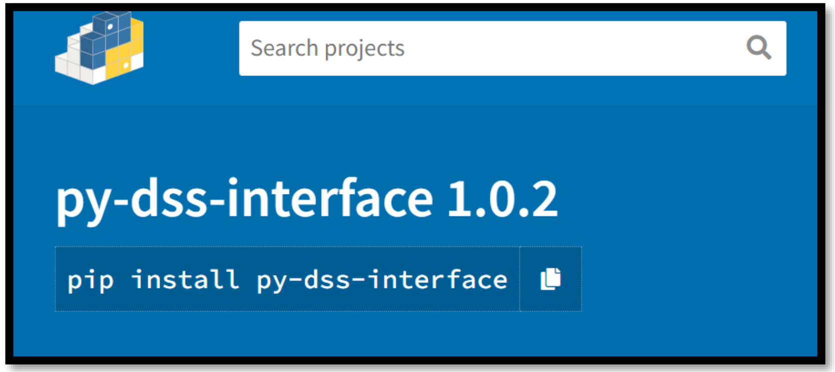
The direct connection shared library is a DLL that implements the same classes, properties, and methods of the OpenDSS-PM COM interface. This alternative was generated to accelerate the In-process co-simulation between OpenDSS and external software when the client software does not support early bindings connection to COM servers/controls.

Normally, high level programming languages do not support early bindings, which make them use late bindings for data exchanging with COM servers. Late bindings procedures add an important overhead to the co-simulation process specially when executing loops.

So, if your programming language does not support early bindings connection with COM servers, this is the library you should use to accelerate your simulations. This library is called OpenDSSDDirect.dll and can be accessed directly without needing to register it into the OS registry.

Python Interface: py-dss-interface

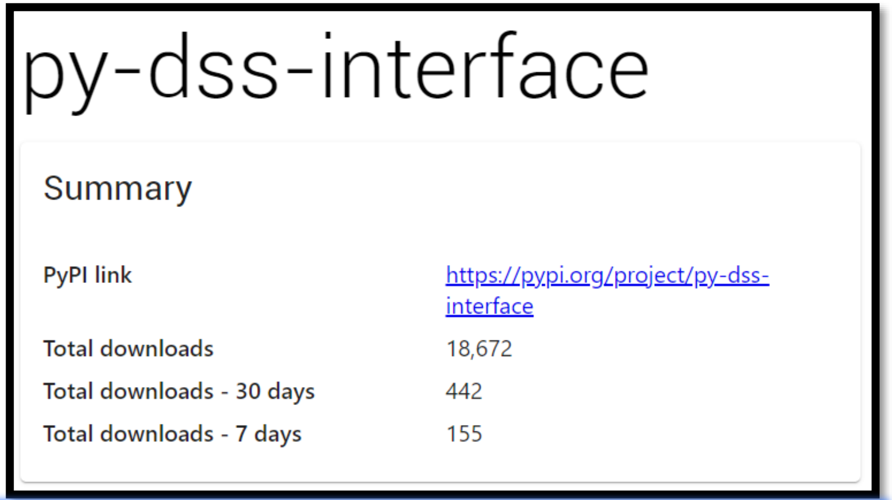
- Python Package
- Uses the official OpenDSS DirectDLL version
  - Version 9.4.0.1 comes with the Package
  - Users can use their OpenDSS version as well



py-dss-interface 1.0.2

```
pip install py-dss-interface
```

- Installing from PyPI
  - *pip install py-dss-interface*



py-dss-interface

Summary

PyPI link	https://pypi.org/project/py-dss-interface
Total downloads	18,672
Total downloads - 30 days	442
Total downloads - 7 days	155

Examples

https://pepy.tech/project/py-dss-interface

Why You Might Want to Write Some Code …

Why You Might Want to Write Some Code …

Create an algorithm not in OpenDSS

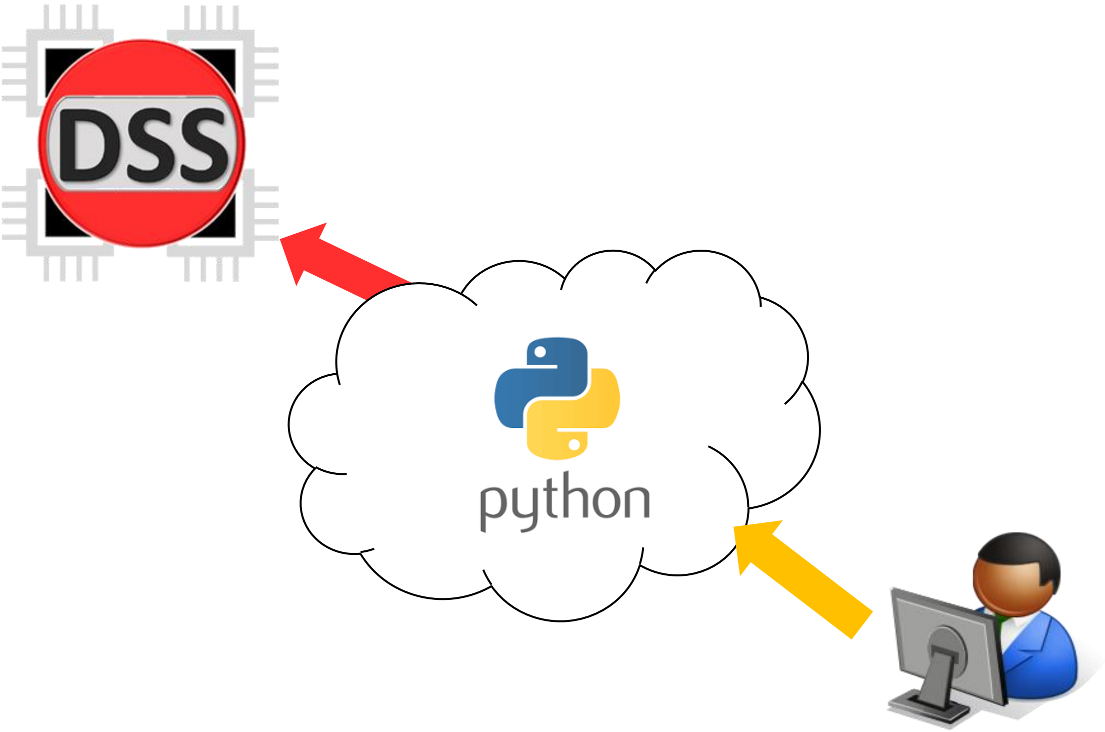
Develop a new device model or control

Automate manual and repetitive tasks

Obtain meaningful results

Basic of Controlling OpenDSS

Basic of Controlling OpenDSS



How to Get Started with Python and OpenDSS?

1. Python Download and Installation
2. PyCharm Download and Installation
3. py-dss-interface YouTube Playlist
4. Examples

1 Load Circuit/Study

- 1-creating_dss_obj.py
- 2-text_method.py

2 Manipulate Circuit, Elements, and Buses properties

- 3-read_circuit_properties.py
- 4_1-manipulate_element_properties.py
- 4_2-manipulate_element_properties.py
- 4_3-manipulate_element_properties.py
- 5-manipulate_bus_properties.py

3 Perform Simulation and Obtain Results

- 6-read_circuit_results.py
- 7-read_elements_results.py
- 8-read_bus_results.py
- 9_1-looping_elements.py
- 9_2-looping_elements.py
- 10-debug.py

4 Examples

- 11-max_voltage_bus.py
- 12-furthest_3pf_bus.py
- 13-max_load_undervoltage.py
- 14-element_properties_dataframe.py

Thank you! Questions?

EPRI 50th

ANNIVERSARY

Together…Shaping the Future of Energy®