Scripting

Scripting allows you to access OpenSim's functionality through the following programming languages:

- The scripting shell in the OpenSim GUI (which is a Jython interpreter embedded in the application)
- Matlab
- Python

In other words, you can access OpenSim's Application Programming Interface without compiling your code in C++.

What's available?

With OpenSim scripting, you can:

- Run tools from setup files or programmatically.
- Perform batch processing of common workflows (e.g., inverse kinematics, computed muscle control, EMG-driven simulation).
- Utilize the OpenSim API without the overhead of learning to program in C++ and setting up a development environment.
- Write "main" programs similar to those written by C++ developers, while taking advantage of the many open-source Matlab/Python packages for data processing, statistics, machine learning, etc.
- Access common SimTK/Simbody numeric types (e.g., Vec3, Vector, Mat33, State, Inertia) and a limited subset of Simbody multibody calculations.
- Access the OpenSim API to create and simulate models.
- Use the Simbody visualizer.

Limitations

- In general, you cannot create new components (e.g., a custom muscle; though there are some exceptions).
- You cannot create plugins for use through the GUI or command-line.
- In Matlab/Python, there's no access to OpenSim's plotter (use Matlab/Python native plotter) or visualizer (use the Simbody visualizer).
- Many SimTK/Simbody classes (that belong to the SimTK namespace and simbody internals) are not available (e.g., integrators).

The sections below outline how to get started with scripting and describe the available functionality.

Next: Common Scripting Commands

Previous: Guide to Using Doxygen

Home: Scripting and Development