Python Library Documentation

Arjun Gupta

January 30, 2018

Contents

0.1 Preface

The goal of this library is to allow users to integrate small python functions and classes into their MOOSApps to make use of the wide range of available python libraries and ease of use of the python language without compromising the speed of C++.

0.2 Class

The library implements a PythonCaller class which has a constructor that initializes the python interpreter, and a destructor that deallocates variables and stops the python interpreter. Once initialized, the *set_program(string program_name)* command must be called to set the python program, and then the *add_funcs(vector[string] funcs)* to initialize the functions that will be used by the PythonCaller class. Once the program and functions are loaded, the functions can be called through the *run(string method, vector[string] sArgs, vector[PyObject*] pArgs, vector[vector[string]] vArgs)* command. Where $sArgs$, $pArgs$, and $vArgs$, are the arguments passed to the function *method*. The arguments are passed to the function with $pArgs$ first, $sArgs$ second, and $vArgs$ last. Where $pArgs$ are meant to hold arguments that are python objects, $sArgs$ are meant to hold arguments that are string objects (can be converted to intergers in python), and $vArgs$ is meant to hold vector inputs (python lists) to the function. The *run* function returns a $PyObject*$ which can then be converted to a C++ style variable for use in the rest of the C++ code.