



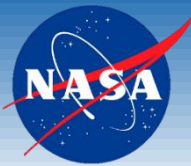
Beginner Interaction and Automation with the Python API and VSPSCRIPT

2022 OpenVSP Workshop
NASA LaRC & NIA, Hampton, VA

August 9-11, 2022

BRANDON LITHERLAND, AST
NASA LANGLEY RESEARCH CENTER
AERONAUTICS SYSTEMS ANALYSIS BRANCH

- Disclaimer -



This is not intended to be a deep-dive into script structure or coding languages.

This presentation and the demonstrations that follow are intended to familiarize new users with the basics of scripting and the APIs.

If you need detailed guidance with an API issue, please post on the OpenVSP Google Group.

Review API Presentation by ESAero






- 2020 OpenVSP Workshop included “[OpenVSP API & MATLAB/Python Integration](#)” by Justin Gravett (ESAero).
- Detailed summary of the API functions, documentation, and how to build.
- Recall the API documentation is located at:
http://openvsp.org/api_docs/latest/
 - Docs written for AngelScript but translation is relatively simple.

API = Application Programming Interface

1

OpenVSP API & MATLAB/Python Integration

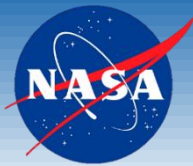


Presented by:
Justin Gravett
justin.gravett@esaero.com
Empirical Systems Aerospace, Inc. (ESAero)

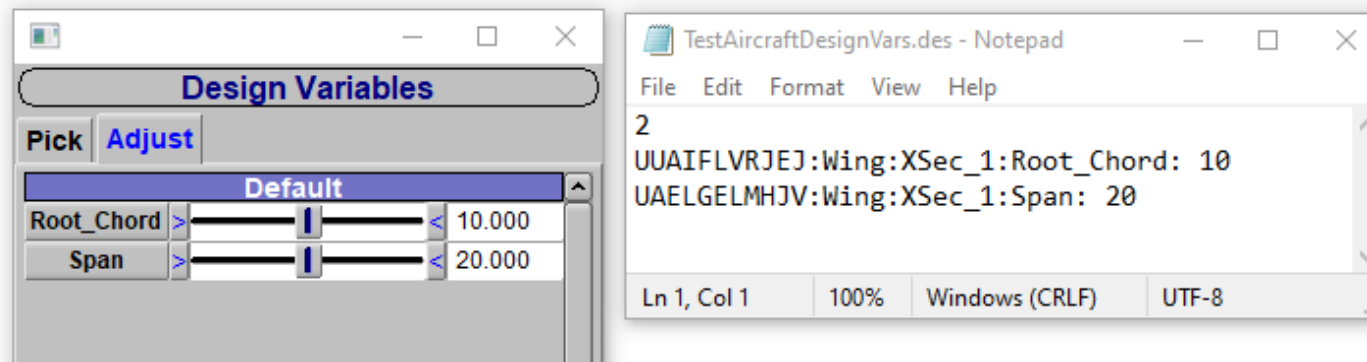
Empirical Systems Aerospace, Inc. www.esaero.com OpenVSP Workshop 2020 Sept. 15th – Sept. 17th

Used with permission from ESAero

Do you need the API?



- *What are you trying to accomplish?*
- Design Variables are one of the fastest/easiest ways of setting parameters.
 - Variable Presets are another method (slightly different).
 - Directly set parameter values via GUI or editing DES/XDDM text file.

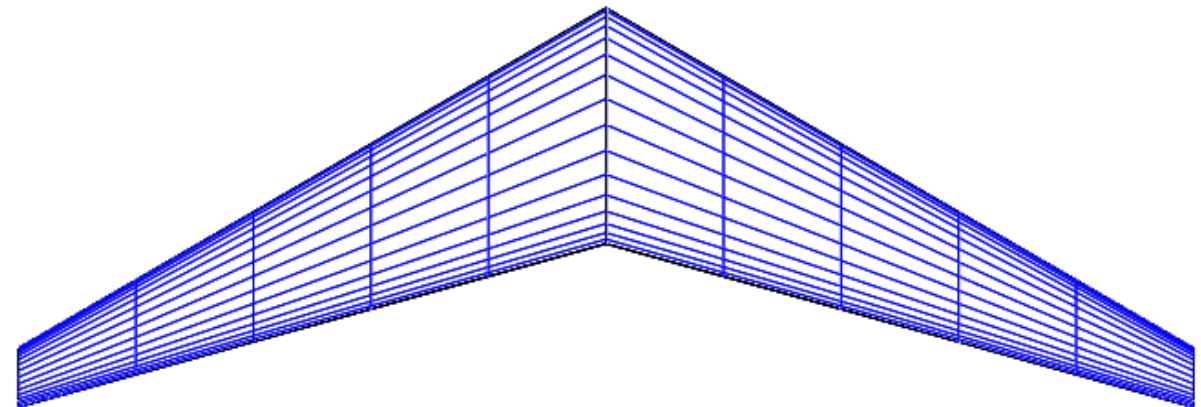
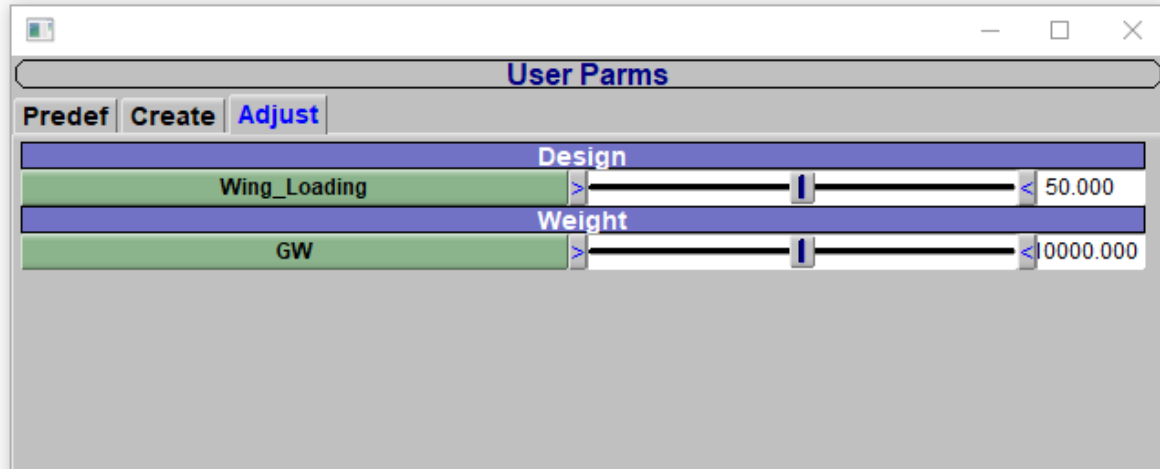


- “Easy” does not imply simplicity. Both Design Vars. and Var. Presets can become quite large or complex depending on the application.
- 2020 VSP Workshop [Design Variables and Variable Presets](#)

Do you need the API?



- Linear or Advanced Links provide parameter connections that can change the way you interact with a model.
 - Combined with User Parameters, you can fundamentally change how a model is parameterized. Sizing a wing by loading and weight, for example.
 - 2020 OpenVSP Workshop “[Parameter Linking](#)” presentation by McDonald.
 - Establish parameter relationships, embed designer intent, reduce dimensionality of design space...
 - May also define intermediate variables within the Advanced Link editor by declaring types. (“double power;” or “int N;” etc...)

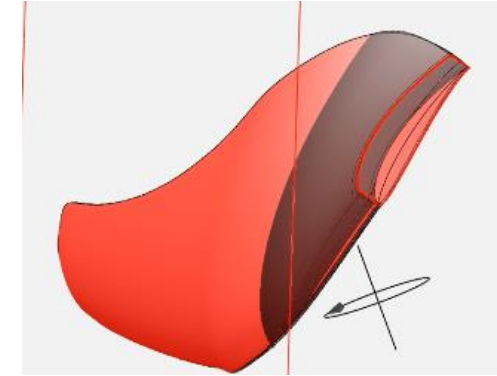


Why use the API?

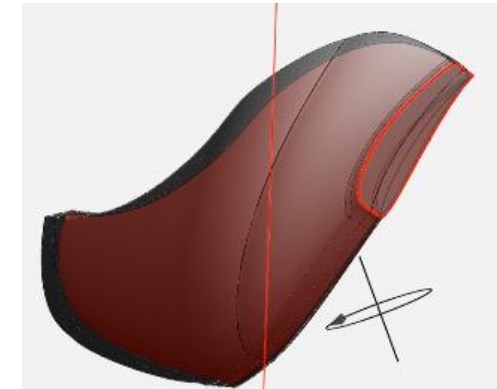


- “Headless” or non-graphical operating systems
- Automation
 - Speed up repetitive or time-consuming tasks (HLP example)
 - Perform analyses and post-processing from memory (sweep VSPAERO run)
 - Perform trade studies and design space explorations
- Interact with other tools and programs
- Testing OpenVSP
- Custom geometries
 - VSP Path/CustomScripts/
- Over 30 example and test scripts provided
 - VSP Path/scripts/

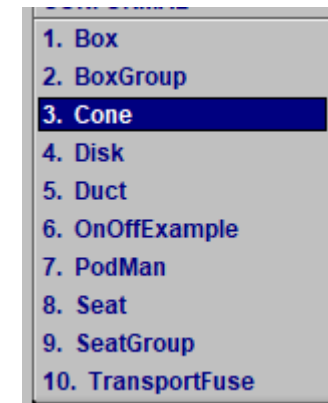
High-Lift Propeller, 80 stations, 4 XSec
~45s altering one section manually



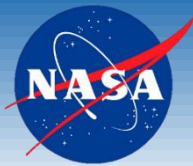
~22s altering all by script



Packaged Custom Geoms



Setting up the API



- VSPSCRIPT comes ready to go.
- Python API is relatively simple to setup.
 - Read the README.md in the /python folder.
 - Ensure that the Python version matches the OpenVSP build downloaded. Or build OpenVSP yourself with different versions. Python environments are your friend!
 - Windows:
 - Run setup.ps1 from Anaconda PowerShell or run “pip install -r requirements-dev.txt” from Python.
 - The PS1 script runs the same steps as the Mac/Linux setup.
 - Mac OS and Linux (Bash):
 1. Execute “conda env create -f ./environment.yml”
 2. Execute “conda activate vsppyttools”
 3. Execute “pip install -r requirements-dev.txt”
 - Test installation:
 - import openvsp as vsp
 - vsp.VSPCheckSetup() or vsp.VSPRenew()
- MATLAB API takes quite a bit more work to build.
 - [OpenVSP API & MATLAB/Python Integration](#).

Windows

[OpenVSP 3.28.0 64-bit Python 3.9](#)

[OpenVSP 3.28.0 64-bit Python 3.6](#)

[OpenVSP 3.28.0 32-bit Python 3.9](#)

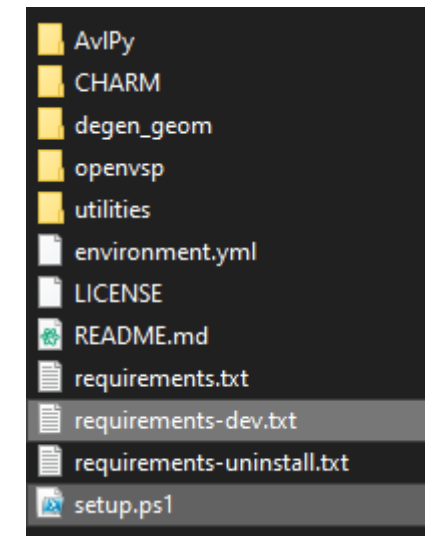
[OpenVSP 3.28.0 32-bit Python 3.6](#)

Mac OS

[OpenVSP 3.28.0 Python 3.9](#)

[OpenVSP 3.28.0 Python 3.6](#)

VSP_Path/python



- Most of the documented API functions are a one-to-one swap between AngelScript and Python.
 - Ex: SetParmVal() becomes openvsp.SetParmVal() or vsp.SetParmVal()
- New users tend to rely on the VSPSCRIPT examples.
 - Relatively simple to modify to achieve your goals.
 - No additional build required.
 - Good place to start to learn “how” to run functions.
- As automation needs become more complex, the Python API becomes far easier to use and more versatile. (In my opinion)
 - Python scripts can be easier to follow for those not well versed in C++.
 - Interact with models, analyses, and results directly from memory without needing to write and parse files. Saves time and resources.
 - Plotting, statistical analysis, post-processing packages in Python but not VSPSCRIPT.

- Refresh the API at the beginning of the script.
 - Also use when looping to clear memory
 - VSPRenew()
 - ClearVSPModel()
 - DeleteAllResults()
- VSPSCRIPT requires declared variable types and the “main” function.
- Good practice to check for errors at the end of the script.

VSPSCRIPT

```
//==== Check For API Errors ====//  
while ( GetNumTotalErrors() > 0 )  
{  
    ErrorObj err = PopLastError();  
    Print( err.GetErrorString() );  
}
```

Python API

```
# Check for errors  
  
num_err = errorMgr.GetNumTotalErrors()  
for i in range(0, num_err):  
    err = errorMgr.PopLastError()  
    print("error = ", err.m_ErrorString)
```

- Some functions expect Geoms only rather than *any* container.
 - User Params exist in a container but cannot be set using the syntax `SetParmVal(userparm_ID, 'Name', 'Group')`.

```
double SetParmVal (const string &in geom_id, const string &in name, const string &in group, double val)
```

- However, you can accomplish setting the Parm via API by searching for the Parm ID...
 1. `up_id = vsp.FindContainer('UserParams', 0)`
 2. `myparamid = vsp.FindParm(up_id, 'Name', 'UserGroup')`
 - Note 'Name' is the name of the parm. Could be preset or custom parm.
 3. `vsp.SetParmVal(myparamid, double val)`
- You may also set the non-Geom parm directly by finding the 10 character ID manually and using that string but the above method is more robust to file changes where the ID can be wiped out.

Creating models from scratch



```
//==== Create some test geometries ====//
Print( string( "--> Generating Geometries" ) );
Print( string( "" ) );

string pod_id = AddGeom( "POD", "" );
string wing_id = AddGeom( "WING", "" );

SetParmVal( wing_id, "X_Rel_Location", "XForm", 2.5 );
SetParmVal( wing_id, "TotalArea", "WingGeom", 25 );

string subsurf_id = AddSubSurf( wing_id, SS_CONTROL, 0 );

Update();

//==== Setup export filenames ====//
string fname_vspaerotests = "TestVSPAero.vsp3";

//==== Save Vehicle to File ====//
Print( "-->Saving vehicle file to: ", false );
Print( fname_vspaerotests, true );
Print( "" );
WriteVSPFile( fname_vspaerotests, SET_ALL );
Print( "COMPLETE\n" );
Update();
```

Add components.

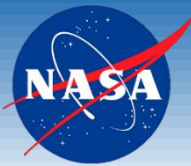
Set some parameters.

Don't forget to Update!

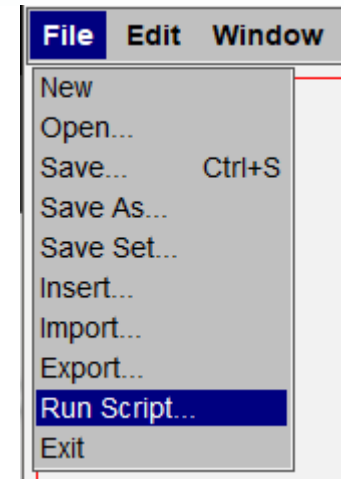
Add Print statements
to check progress and
operation.

Save the model for later.

Running VSPSCRIPT



- Run scripts from the GUI by choosing File > Run Script... from the menu bar.
- Run scripts from command line with:
 - `vspscript.exe -script scriptname.vspscript`
- Interacting with existing models:
 - Run the script from the GUI after opening or creating model.
 - Read a file from within the script.
 - `ReadVSPFile("modelName.vsp3")`
 - `Update()`
 - Pass the script a filename from the command line.
 - `vspscript.exe -script scriptname.vspscript
modelName.vsp3`
 - Note: file paths may be used to run from different directories.
 - `"Path/to/VSP/vspscript.exe" -script "Path/to/script/scriptname.vspscript"`



Other tips



- The /matlab folder has MATLAB scripts that interact with VSP files or data.
 - Plotting DegenGeom shape, Parasite Drag tool data, Bezier airfoils...
- Recall custom Geoms may be built with AngelScript as VSPPART files.
 - See /CustomScripts directory for examples.
 - These are loaded into your Geom Browser dropdown.
 - Ex) [OpenVSP Custom Split-Flow Turbofan Engine Model](#) presentation (2021 VSP Workshop).
- The /python folder has several useful Python API functions besides OpenVSP.
 - CHARM automation
 - AvIPy
 - degen_geom
 - utilities
 - Atmosphere, plotting, runners, rotor calculations, unit conversion, and more.

How to troubleshoot issues

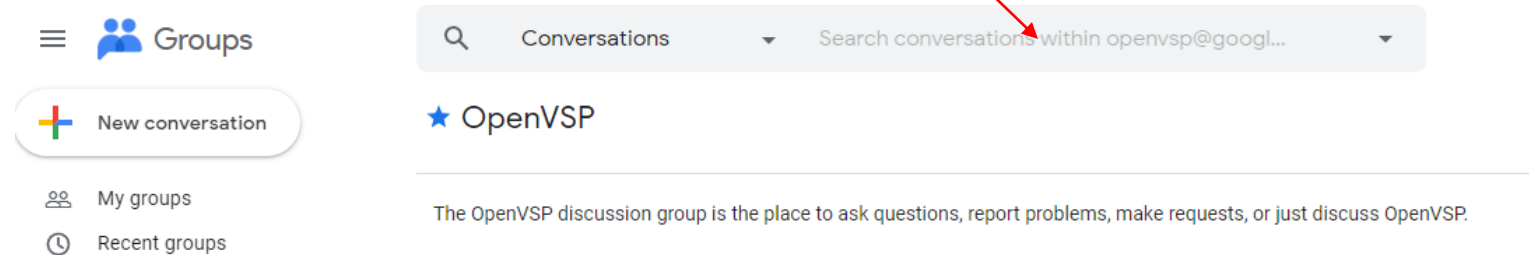


- Check the API documentation

OpenVSPAPI 3.28.0

Documentation for the OpenVSP API

- Double check if there are example scripts to get you started.
- Ask the Google Group (search first, please)



- Check syntax errors...
- DLL load error?
 - Make sure that the Python version matches the OpenVSP compiled version.



Thank you!

Questions?

Demo to follow...