

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

2

- Introduction
- C++ API
- AngelScript API
- Python API
- MATLAB API Overview
- MATLAB API Build Instructions*

www.esaero.com

*Recommended: Building OpenVSP from Source Presentation





Introduction

- API: Application Programming Interface
- Languages:
 - C++
 - AngelScript (close to C++)
 - Python
 - MATLAB (experimental)
- What is it used for?
 - Headless operating systems
 - Automation
 - Integration with other codes
 - Test suites
 - Custom Geoms
- How to learn:
 - Old way: study the source code
 - Examples
 - New: Doxygen documentation of functions and classes
 - Workshop presentations

8







API Functions



- API Documentation: <u>http://openvsp.org/api_docs/latest/</u>
 - Available for direct download
- Analysis and Results Manager:
 - Generic functions to setup analysis inputs, execute analyses, and collect results
 - Strings used to identify available analysis types
 - i.e. "VSPAEROComputeGeometry"
 - Preferred approach for running analyses when available
 - Use "ListAnalysis()" to identify available analysis strings
 - Used "PrintAnalysisInputs("analysis") to identify available inputs
 - Arrays used for analysis input and result data
- Parms Through the API
 - Generic functions for controlling Parms
 - Values cast to double type in API
 - Parms identified by ID or Container, Group, and Name
- Identifying Parm Info:
 - OpenVSP *.vsp3 file
 - Click and drag GUI device to text editor
 - Click on the GUI Parm button

		_	×
\square	Parm		
Info Link	s Adv Links		
	Parm Info		
Name:	Y_Rel_Location		
Group:	XForm		
Desc:	Y Location Relative to Parer	nt	
Min Val:	-100000000000.00000		
Curr Val:	0.00000		
Max Val:	100000000000.00000		
	-		



C++ API

- Basis for all other supported languages OpenVSP's primary language is C++
- Functions written in VSP_Geom_API (geom_api project)
- Exposes core functionality to higher level abstraction
- Used for API testing
 - Test suites for Parasite Drag, Mass Properties, VSPAERO, and more
 - src -> geom_api folder of repository (apitest project)
 - CFD Mesh test suite in progress
 - Run prior to each release to help catch bugs
 - Intended to be included with continuous integration builds, but currently too time consuming

AngelScript API

6

EISIAERO

- Very similar to C++
 - Must declare variable types
 - "main" function required
- Same language used in Advanced Links and for Custom Geom scripts
- *.vspscript file extension required
- C++ API functions, classes, and enums are registered through AngelScript
- 30+ example scripts provided in scripts directory
 - Demonstrate how to use various API functionality
 - Examples of API automation
 - Most advanced: Master_VSP_VV_Script.vspscript
- 2 ways to run *vspscript files:
 - 1. OpenVSP GUI: File -> Run Script...
 - 2. VSPSCRIPT executable:
 - Run through command terminal
 - Example: vspscript -script "scripts/CpSlicer.vspscript"



	_		×
Expo	ort Cust	tom Sc	ript 🔵
	Sustom S	Scripts	
Box			
Cone	p		
Disk			
Duct			
OnOffExa	ample		
PodMan			
SeatGrou	a		
Transpor	tFuse		
Save T	o File	Canc	el

www.esaero.com



Python API

- SWIG: Simplified Wrapper and Interface Generator
 Wraps C++ API for Python
- Python version when importing OpenVSP API must match version used by SWIG
 - Python 3.6: latest version used with OpenVSP distribution
 - Must compile OpenVSP yourself if alternative Python version is desired
- *Windows 3.21.2 initially released without Python API – fixed and updated



Python API

- Installation instructions available in python folder of OpenVSP distribution – see README
 - <u>https://github.com/OpenVSP/OpenVSP/tree/master/src</u>
 <u>/python_api/packages</u>
 - Open a command terminal in the same directory
 - Recommended (Windows) use the "dev" API installation version
- Test installation
 - Open command terminal and type:
 - "python"
 - "import openvsp as vsp"
 - "vsp.VSPCheckSetup()" or "vsp.VSPRenew()"



Python API – Common Errors

- DLL load error?
 - Check your python version matches version compiled with OpenVSP
- Missing VSPAERO solver, viewer, or slicer?
 - Copy executables to Python directory

VSPAERO solver not found. VSPAERO viewer not found. VSPAERO slicer not found.





MATLAB API

- Advanced and experimental
 - Not recommended for new API users
 - Requires existing software development skills
- Requires unreleased version of SWIG

ESAero Support

- Integrated MATLAB API with PANTHER toolset in Phase I SBIR
- Minor modifications made to OpenVSP – released publicly
 - Duplicated test suite to verify functionality
 - Tried to keep the build process as simple as possible



NASA SBIR PHASE I - UAM Disturbance Rejection in Conceptual Design



Software – Python API

Install Anaconda

Anaconda Version Archive

- Version 5.2.0 Includes
 Python 3.6, compatible with
 OpenVSP 3.21.2
- Latest Version
 - Only use this version if compiling OpenVSP yourself
- Add to PATH
 - Recommend through installation option
 - Could manually adding to system PATH

Anaconda3 2020.07 (64-	bit) Setup		10		×
O ANACONDA.	Advanced Insta Customize how A	allation Options	tes with Windows	5	
Advanced Options					
Add Anaconda3 t	o the system PATH	environment vari	able		
Not recommended. I menu and select "An Anaconda get found cause problems requ Register Anacond This will allow other p PyCharm, Wing IDE, detect Anaconda as	nstead, open Anac aconda (64-bit)". Ti before previously i iring you to uninsta la3 as the system P programs, such as P PyDev, and MSI bit the primary Python	onda3 with the W his "add to PATH" Installed software II and reinstall And ython 3.8 ython Tools for V hary packages, to 3.8 on the syste	indows Start option makes , but may aconda. isual Studio o automatically m.		
Anaconda, Inc		< Back	Install	Can	cel



SWIG - Python

- Only needed if compiling OpenVSP
- SWIG Simplified Wrapper and Interface Generator
 - Official distribution
 - Used to provide python interface to C-API
 - http://www.swig.org/download.html
 - Recent CMake issue with OpenVSP finding 4.0.2
 could use 4.0.1 instead



SWIG - MATLAB

- SWIG MATLAB
 - Unofficial SWIG build with MATLAB bindings
 - https://github.com/jaeandersson/swig.git
- User must compile OpenVSP
- Git required
 - Sourcetree recommended Git GUI
- MinGW, MSYS, & PCRE used

Build Process – Overview



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is working

Build Process – Overview



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build so ution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MA TLAB to verify everything is working

These are the most challenging steps

Empirical Systems Aerospace, Inc.

www.esaero.com



1. Clone the swig-matlab repo & checkout matlab-customdoc

- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that ye
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is wor

	Go to file ↓ Code ↓
Clo	one with HTTPS ?
Use	Git or checkout with SVN using the web URL.
h	ttps://github.com/jaeandersson/swig.g
Ť	Open with GitHub Desktop
3	Download ZIP

WORKSPACE	All Branches	Show Remote Branches	Ancestor Order 🐣				Jump to:	
	Graph		Description		Date	Author	Commit 🔨	
File Status	•	🕑 origin/matlab-customdoc	Support for maxargs/maxargs		1 Aug 2019 6:14	Joris Gillis <joris.gi< th=""><th>82ca29bc</th></joris.gi<>	82ca29bc	
History	•	Py37 adaptions			4 Nov 2018 11:52	Joris Gillis <joris.gi< th=""><th>9185f9cf7</th></joris.gi<>	9185f9cf7	
Search		• 🚺 matlab-customdoc 🔰 🚺	ESA_origin/matlab-customdoc	Enlarge erro	21 Aug 2018 13:17	Joris Gillis <joris.gi< th=""><th>c0c7b554</th></joris.gi<>	c0c7b554	
	l I 🛉	Fix extending class in subpack	age.		21 Aug 2018 13:17	Yuriy Kozlov <yuriy< th=""><th>247f6a64</th></yuriy<>	247f6a64	
✓ Å ۶ BRANCHES	•	Hotfix for octave 'delete' issue	S		9 Jul 2018 6:35	Joris Gillis <joris.gi< th=""><th>0fc41174</th></joris.gi<>	0fc41174	
8	•	Workaround for octave bug #53844			19 May 2018 1:52	Joris Gillis <joris.gi< td=""><td>c7689b2a</td></joris.gi<>	c7689b2a	
master	l 🛉	Using method for clearing swi	15 Apr 2018 14:34	Joris Gillis <joris.gi< th=""><th>60ca96a6</th></joris.gi<>	60ca96a6			
O matlab-customdoc	l 🛉	Removed unused variable			8 Nov 2017 11:35	Joel Andersson <j.< th=""><th>8ff939ef2</th></j.<>	8ff939ef2	
	•	Adding %matlabcode static. %	6matlabcode class		6 Nov 2017 7:52	Joris Gillis: u00523	167d431(
> 🏷 TAGS	SWIG-MATLAB							
	Ori	ain• https•//aithub	o com/iaeanderssc	n/swia a	it			
ESA_origin		gini napol/giniac	am de e	n,omg.g				
	Bla	inch: matiab-cusi	OMOOC					
2 oligin		ESAero branc	h off of 0fc411742	bdfb36c3	3180d5b671	a1da92766	af90e	
> 🗞 STASHES	Two commits charry nicked:							
	Use	e Latest Git SHA:	: 82ca29bc988979	92856a50	198c5903fb	1688564217		

Empirical Systems Aerospace, Inc.

www.esaero.com

17



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to I sin sin
- 5. Compile OpenVSP (Build solution, ←→
- 6. Delete duplicate SwigClear(self) fu
- 7. Build some test scripts in MATLAB

716	Simpl	ified W	rapper and	× 0	Search · swig-matlab	× 🛤	VSP - Matlab	- Interfac 🗙	0	Getting started on Win $$ ×	+		-		\times
-	\rightarrow	C	(i) File	C:/Repo	_swig-matlab/Doo	/Manual/	Windows.htm	nl#Windows_	swig_	exe		☆ 0	S	N	:
3.	3.1] If you neces envir	Build a want sary to onmer	ing swig t to replica o use the s nt. Normal	g.exe on ate the bu supplied lly this is	Windows hild of swig.exe to swig.exe. This in s not needed, so r	hat comes formation nost peop	with the dov is provided will want	wnload, follo for those tha to ignore this	ow th at wa s sect	e MinGW instructions nt to modify the SWIC tion.	below. source	This is not code in a	Wind	ows	•

3.3.1.1 Building swig.exe using MinGW and MSYS

The short abbreviated instructions follow ...

- Install MinGW and MSYS from the <u>MinGW</u> site. This provides a Unix environment on Windows.
- Follow the usual Unix instructions in the README file in the SWIG root directory to build swig.exe from the MinGW command prompt.

The step by step instructions to download and install MinGW and MSYS, then download and build the latest version of SWIG from Github follow... Note that the instructions for obtaining SWIG from Github are also online at <u>SWIG Bleeding Edge</u>.

Pitfall note: Execute the steps in the order shown and don't use spaces in path names. In fact it is best to use the default installation directories.

Documentation in source repository: Doc\Manual\Windows.html

Instructions exist to build using

- MinGW & MSYS ESAero Tested
- Cygwin
- VisualStudio IDE (challenging)

 Download the following packages from the <u>MinGW download page</u> or <u>MinGW SourceForge download page</u>. Note that at the time of writing, the majority of these are in the Current release list and some are in the Snapshot or Previous release list.

- MinGW-3.1.0-1.exe
- MSYS-1.0.11-2004.04.30-1.exe
- msysDTK-1.0.1.exe
- bison-2.0-MSYS.tar.gz
- msys-autoconf-2.59.tar.bz2
- msys-automake-1.8.2.tar.bz2
- 2. Install MinGW-3.1.0-1.exe (C:\MinGW is default location.)
- Install MSYS-1.0.11-2004.04.30-1.exe. Make sure you install it on the same windows drive letter as MinGW (C:\msys\1.0 is default). In the post install script.
 - ult). In the post install script,
 - Answer y to the "do you wish to continue with the post install?"
 Answer y to the "do you have MinGW installed?"
 - Answer y to the "do you have MinGW installed?"
 Type in the folder in which you installed MinGW (C:/MinGW is default)
- Type in the folder in which you installed MinGW (C:/MinGW is default)
 Install msysDTK-1.0.1.exe to the same folder that you installed MSYS (C:/msys\1.0 is default).
- 5. Copy the following to the MSYS install folder (C:\msys\1.0 is default):
 - msys-automake-1.8.2.tar.bz2
 - msys-autoconf-2.59.tar.bz2
 - bison-2.0-MSYS.tar.gz
- Start the MSYS command prompt and execute:

Empirical Systems Aerospace, Inc.



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe

3. Create a 'matlab_api' (use the 'python_api' as a template)

<u>A</u> Doint th	A VSD CN	$\Lambda \Lambda \mu$	(Lonti	one to l	ook tor the swid eve	s tha	t \	Edit environment variable	×
🏇 MinGW Installation Manager					-)		
Installation Package Settings						Help	6		
Basic Setup	Package	Class	Installed Version	Repository Version	Description		1	%USERPROFILE%\AppData\Local\Microsoft\WindowsApps	New
All Packages		hin		2013072300	An MSYS Installation for MinGW Developers (meta)			C:\Program Files\CMake\bin	
	mingwaeveloper tookt	bin		2013072200	A Basic MinGW Installation			C:\Program Files\doxygen\bin	Edit
	mingw32-gcc-ada	bin		6.3.0-1	The GNU Ada Compiler			C:\MinGW\bin	Lan
	S mingw32-gcc-fortran	bin		6.3.0-1	The GNU FORTRAN Compiler				-
	mingw32-gcc-g++	bin		6.3.0-1	The GNU C++ Compiler			C:\MinGW\msys\1.0\bin	Browse
	mingw32-gcc-objc	bin		6.3.0-1	The GNU Objective-C Compiler				
	msys-base	bin		2013072300	A Basic MSYS Installation (meta)				
									Delete
									-
		_					- 1		
	General Description Depe	endencies	Installed Files Ve	rsions					Marialla
	An MSYS Installation for MinGW Developers (meta)								wove up
	This meta package contain	is the co	mponents necessary	to create a relatively	complete MSYS installation suitable for most MinGW develope	ers' needs. It			
	native win32 (MinGW) progra	ams. It i	well as the autotools ncludes m4, perl, vim	(autoconf, automake, , patch, cvs, guile, mkt	Interior (interior)	and building			Move Down
	networking clients for ssh, rs	sh, telnet	t, and ftp protocols.						
	Note that, for certain esoter	ric reason	is, some of the pack	ages installed by this n	neta package are MSYS packages, and install, as expected, in	the / == /usr			
	technically configured for Min	nGW, and	d are installed into th	e /mingw directory (that	t is, into C:\MinGW or whatever you've chosen as the "root" MinG	GW path).			
									Edit text
								Add MinCW and	-
									-
		Mi	nGW I	nstalla	tion Packages			MSYS to Path	
				instanta	lion i achages				
								OK	Cancel

- No need to install bison, automake, and autoconf separately (already included)
- PCRE (pcre-8.44.tar.gz): <u>ftp://ftp.pcre.org/pub/pcre/</u> or <u>https://ftp.pcre.org/pub/pcre/</u>
 - Follow instructions to extract in Swig source repository and run "Tools/pcrebuild.sh"

Empirical Systems Aerospace, Inc.

www.esaero.com



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)

4.	Point the \	Command Prompt	_		×
5.	Compile C	checking dependency style of gcc gcc3 checking for g++ g++ checking whether we are using the GNU C++ compiler ves			^
6.	Delete dup	checking whether g++ accepts -g yes checking dependency style of g++ gcc3 checking whether gcc and cc understand cc and co together was			
7.	Build some	checking whether to enable PCRE support yes checking for pore-config /c/Users/justin.gravett/Documents/swig/pcre/pcre-swig-install/bin/pcre-config checking for pcre-config /c/Users/justin.gravett/Documents/swig/pcre/pcre-swig-install/bin/pcre-config checking for pcre-config /c/Users/justin.gravett/Documents/swig/pcre/pcre-swig-install/bin/pcre-config checking for pcre-config /c/Users/justin.gravett/Documents/swig/pcre/pcre-swig-install/bin/pcre-config checking for pcre-config /c/Users/justin.gravett/Documents/swig/pcre/pcre-swig-install/bin/pcre-config checking whether to enable ccache-swig yes	or C ocre-c	onfig	
		Checking packages required for SWIG developers. Note : None of the following packages are required for users to compile and install SWIG from the distri	outed '	tarball	L
		checking for bison bison -y			
		Checking for installed target languages and other information in order to compile and run the examples and test-suite invoked by 'make check'. Note : None of the following packages are required for users to compile and install SWIG from the distrib	outed	tarball	L
		configure: line 5873: syntax error near unexpected token `fi' configure: line 5873: `fi' 			~

Solution to error after calling "bash configure":

- 1. Delete configure file
- 2. Rerun "bash autogen.sh"
- 3. Open configure delete lines 5773 through 6027

Empirical Systems Aerospace, Inc.

www.esaero.com



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is working

🗾 C	:\OpenVSP\repo\src\CMakeLists.txt - Notepad++	— —	\times
File	Edit Search View Encoding Language Settings Tools Macro Run Plugins	Window ?	х
6) 🖯 🛍 🗟 🐚 📥 🏑 🐚 🌔 Ə 🧲 🏙 🍢 🔍 🔍 🖽 🔁 🎫 1 📗	i 🖉 💹 🖉 💼 💌 📄 📄	
🔚 CN	lakeLists.txt 🔀		
73	ADD_SUBDIRECTORY(·xmlvsp·)		~
74	ADD_SUBDIRECTORY(·geom_core·)	Talla CNAN/C whata	
75	ADD_SUBDIRECTORY(.cfd_mesh.)		
76		the second second second second	
77	IF (· NOT ·VSP_NO_GRAPHICS·)	the source code is for	
78	<pre>ADD_SUBDIRECTORY(·gui_and_draw·)</pre>		
79	ENDIF()	the MAILAB API is	
80			
81	ADD_SUBDIRECTORY(·geom_api·)		
82	ADD_SUBDIRECTORY(.vsp.)		
83	ADD_SUBDIRECTORY(·python_api·)		
84	ADD_SUBDIRECTORY(+matlab_api+)		
85			
86	ADD_SUBDIRECTORY(·vsp_aero·)		
87			
88	SET (· CPACK_GENERATOR · ZIP ·)		
89	SET (· CPACK_PACKAGE_NAME · OpenVSP ·)		
90	SET (· CPACK_PACKAGE_VERSION_MAJOR · \$ {VSPVER_MAJOR }	•)	
91	SET (· CPACK_PACKAGE_VERSION_MINOR · \$ { VSPVER_MINOR }	•)	
92	SET (· CPACK PACKAGE VERSION PATCH · \$ { VSPVER PATCH }	•)	
0.2			



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is working

> This PC > OS (C:) > OpenVSP > repo > src > matlab_api							
Name	Date modified	Туре	Size				
1 APITestMain.m	9/12/2019 1:35 PM	MATLAB Code	2 KB				
Le APITestSuite_test.m	9/12/2019 1:35 PM	MATLAB Code	38 KB				
APITestSuiteVSPAERO_test.m	9/12/2019 1:35 PM	MATLAB Code	84 KB				
慉 assert_delta.m	9/12/2019 1:35 PM	MATLAB Code	1 KB				
📔 CMakeLists.txt	9/12/2019 1:35 PM	TXT File	5 KB				
📔 Matlabdef.def	9/12/2019 1:35 PM	DEF File	1 KB				
🚺 vsp.i	9/12/2019 1:35 PM	ideaMaker Print File	1 KB				
🚺 vsp_common.i	9/12/2019 1:35 PM	ideaMaker Print File	2 KB				
🚺 vsp_g.i	9/12/2019 1:35 PM	ideaMaker Print File	1 KB				

Copy *.i files from python API example



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is working

Key changes to VSP source

- Explicit control of VSPAERO path
- Reference to Matrix.h



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in MATLAB to verify everything is working

CMake 3.12.1 - C:/OpenVSP/build_Release_vsp_vs2015	- 🗆 X
File Tools Options Help	
Where is the source code: C:/OpenVSP/repo	Browse Source
Where to build the binaries: C:/OpenVSP/build_Release_vsp_vs2015	Srowse Build
Search:	Grouped Advanced 🔂 Add Entry 🗱 Remove Entry
Name	Value
GLEW_LIBRARY GLFONT2_INCLUDE_DIR GLM_INCLUDE_DIR GUI_AND_DRAW_INCLUDE_DIR LIBIGES_INCLUDE_DIR LIBIGES_LIBRARIES Matlab_ROOT_DIR NANOFLANN_INCLUDE_DIR SCREENS_INCLUDE_DIR SIXSERIES_INCLUDE_DIR	C:/OpenVSP/build_Release_libraries_vs2015/GLEW-prefix/lib/g C:/OpenVSP/repo/src/external/glfont2 C:/OpenVSP/build_Release_libraries_vs2015/GLM-prefix/src/GL C:/OpenVSP/repo/src/gui_and_draw C:/OpenVSP/build_Release_libraries_vs2015/LIBIGES-prefix/inc C:/OpenVSP/build_Release_libraries_vs2015/LIBIGES-prefix/bi C:/Program Files/MATLAB/R2018a C:/OpenVSP/repo/src/external/nanoflann C:/OpenVSP/build_Release_vsp_vs2015/src/flk_screens C:/OpenVSP/repo/src/external/sixseries
SWIG_EXECUTABLE	C:/Repo_swig-matlab/swig.exe
TRITRI_INCLUDE_DIR UTIL_INCLUDE_DIR VSP_GRAPHIC_INCLUDE_DIR WAVEDRAGEL_INCLUDE_DIR XMLVSP_INCLUDE_DIR	C:/OpenVSP/repo/src/external/tritri C:/OpenVSP/repo/src/util C:/OpenVSP/repo/src/util C:/OpenVSP/repo/src/vsp_graphic/include C:/OpenVSP/repo/src/external/wavedragEL C:/OpenVSP/repo/src/xmlvsp



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in matlab to verify everything is working

If you are successful....

	Press Configure to update and display new values in red, then press Generate to generate selected build files.									
Configure	Generate	Open Project	Current Generator: Visual Studio 14 2015 Win64							
PYTHON_LIB PYTHON INC	RARIES: LUDE PATH:	C:/ProgramI C:/ProgramI)ata/Anaconda3/libs/python37.lib)ata/Anaconda3/include	^						
SWIG & MATLAB swig.exe: Matlab_INC Matlab_LIB Matlab_MEX	Found, MATI C:/Repo_swi LUDE_DIRS: RARIES: LIBRARY:	LAB MEX will ig-matlab/sv C:/Program C:/Program C:/Program	<pre>be compiled. ig.exe Files/MATLAB/R2018a/extern/include Files/MATLAB/R2018a/extern/lib/win64/microsoft/libmex.lib;C:/Program Files, Files/MATLAB/R2018a/extern/lib/win64/microsoft/libmex.lib</pre>	/MJ						
Matlab_MX C:/Program Fil Found OpenMP_C Found OpenMP_C Found OpenMP:	LIBRARY: es/MATLAB/H : -openmp XX: -openmp TRUE	C:/Program R2018a/exter	Files/MATLAB/R2018a/extern/lib/win64/microsoft/libmx.lib n/lib/win64/microsoft/libmex.lib;C:/Program Files/MATLAB/R2018a/extern/lib/	/w:						
Configuring do	ne			>						



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m

Build Administrator: Windows PowerShell	_	×
 Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/+vsp/Y_PROJ.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/+vsp/Z_DIR.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/vspMEX.mexw64 Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/swigGet.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/SwigGet.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/SwigGet.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/SwigGet.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/APITestMain.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/APITestMain.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/APITestSuite_test.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/matlab_api/APITestSuite_test.m Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.msvcp140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.concrt140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.vcomp140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.vcomp140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.vcomp140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.vcomp140.dll Up-to-date: C:/AeroTools/OpenVSP_DevBuild_Release/.vspaero.exe Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.vspaero.exe Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.vspaero.exe Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.vspaero.exe Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.vspilcer.exe Finalling: C:/AeroTools/OpenVSP_DevBuild_Release/.vspilcer.exe Finalling: C:/AeroTools/OpenVSP_DevBuild_Release/.vspilcer.exe Installing: C:/AeroTools/OpenVSP_DevBuild_Release/.vspilcer.exe <		^
Build succeeded. 0 Warning(s) 0 Error(s)		~

7.



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m
- 7. Build some test scripts in matlab to verify everything is working

Current Folder	🗩 📝 Editor - C:\AeroTools\OpenVSP_DevBuild_Release\matlab_api\SwigRef.m
Current Folder Name * ApiTestMain.m APITestSuite_test.m APITestSuiteVSPAERO_test.m SwigGet.m SwigGet.m SwigGet.m SwigMem.m VspMEX.mexw64	SwigRef.m x + 23 - self.swigPtr = ptr; 24 - end 25 - function SwigClear(self) 26 - self.swigPtr = []; 27 - end 28 - function SwigClear(self) 29 - self.swigPtr = []; 30 - end
	31 function ptr = SwigGet(self) 32 - ptr = self.swigPtr; 33 - end Command Window
	Error Details: Error: File: SwigRef.m Line: 25 Column: 14 'SwigClear' is already defined as a method in the 'SwigRef' class. Error in APITestSuiteVSPAERO_test>setup (line 14) assert(strcmp(vsp.ErrorMgrSingleton.getInstance.GetLastError().GetErrorString(),'No Error'),vsp.ErrorMgrSingleton.getInstance.PopLastError().GetErrorString());
al Systems Aerospace, Inc.	www.esaero.com OpenVSP Workshop 2020 Sept. 15 th –



- 1. Clone the swig-matlab repo & checkout matlab-customdoc
- 2. Compile swig-matlab from source and generate a swig.exe
- 3. Create a 'matlab_api' (use the 'python_api' as a template)
- 4. Point the VSP CMAKE options to look for the swig.exe that you just created
- 5. Compile OpenVSP (Build solution, Build INSTALL)
- 6. Delete duplicate SwigClear(self) function in SwigRef.m

<pre>% ·APITestMain clc clear ·all close ·all</pre>	APITestMain.m		APITestSuiteTest.m
%%·setup·global global·TEST_TOI TEST_TOL·=·le-3	ls ;; ;;····\$·used·in·functions·for·delta·assessmer	15 10 15 10 10	<pre>%% -Fresh -Fixture -Functions function -setup(~) - % -do -not -change -function -namevsp.VSPCheckSetup();vsp.VSPRenew(); end</pre>
global ∙m_vspfna m_vspfname_for_	ame_for_vspaerotests; vspaerotests.=.'apitest_TestVSPAero.vsp3';	11 12 13	function teardown (~) ··* do ·not ·change ·function ·name end
global m_vspfna m_vspfname_for_ global VSPAERO VSPAERO_PATH =	<pre>ame_for_vspaerofunctionalitytests; vspaerofunctionalitytests =- 'apitest_TestVSE PATH [pwd '\\'];</pre>	24 PAeroFunctionality.vsp3'; 16 17 18 19 00 00	<pre>%% void .APITestSuite::CheckSetup() function .CheckSetup test (~)% .fprintf(.'APITestSuite::CheckSetup() \n' .);</pre>
%% ·Execute ·test % ·% ·Simple ·run % ·results ·= ·rur % ·rt ·= ·table (re	:s •syntax 1(APITestSuiteTest); esults)	21 22 23 24 25	<pre>end</pre>
<pre>% ·Complex ·runne import ·matlab.u import ·matlab.u % ·import ·matlak % ·import ·matlak</pre>	er with progress display unittest.TestRunner unittest.TestSuite o.unittest.plugins.TestRunProgressPlugin o.unittest.plugins.FailureDiagnosticsPlugin	26 27 28 29 30 31 32	<pre>%% -void ·APITestSuite :: CreateGeometry () function -CreateGeometry_test (~)% -fprintf (.'APITestSuite :: CreateGeometry () \n' .);types := .vsp.GetGeomTypes (.);</pre>
%.Create.silent runner.=.TestRu %.runner.addPlu %.runner.addPlu	; ·test ·runner ·and ·add ·plug-in ·to ·display ·prog inner .withTextOutput; igin (TestRunProgressPlugin .withVerbosity (mat] igin (FailureDiagnosticsPlugin)	33 gress 34 14 Lab.unittest.Verbosity.Detailed)) 36 37 38	<pre>assert(numel(types)~=0)%.fprintf(.'\t[ndx]\t%20s\t[geom_id]\t%25s\t[num_geoms]\n','[geom_type]','[geom_type]'.);n_extra_geom_sv=.0;for.i_geom_type:=-1:numel(types)</pre>
<pre>% .Run .general .t suite .= .TestSui result .= .run(ru</pre>	<pre>tests ite.fromFile('APITestSuite_test.m'); anner,suite)</pre>	39 40 41	<pre></pre>
<pre>% ·Run ·VSPAERO ·u suiteVSPAFRO ·=</pre>	nit.tests TestSuite fromFile('APITestSuiteVSPAFRO test	- m') ·	

Empirical Systems Aerospace, Inc.

result ·= · run (runner, suiteVSPAERO([7]))



Questions?

www.esaero.com

OpenVSP Workshop 2020 Sept. 15th – Sept. 17th