

Scripting and API

J.R. Gloudemans

Why

- create and analyze geometry without GUI
- automate repetitive tasks
- more easily integrate VSP with other codes
- optimization
- customize results

Scripting vs API

- scripting uses Angelscript
- API uses C++
- no compiling required for scripts
- scripting can be called during execution
- greater flexibility with API

Scripting

- all scripting functions and enums are in Wiki
- name script files with “.as” suffix
- must contain one function named “void main()”
- use Run->Run Script to execute from GUI
- use -script <filename.as> from batch mode
- simple functions
- memory management and pointers are handled in VSP

AngelScript

- open source, cross platform, and mature (2005)
- C/C++ syntax
- OK speed - function calls 2x to 5x slower than native C++
- includes math, array, and string classes
- script parse error are displayed in console window
- links to the Angelscript manual in Wiki

Error Stack

- single persistent error stack
- query for errors at any time
- ErrorObj contain error code and description
- optional query after any line of code:

```
string pod_id = AddGeom("POD", "");  
if ( GetErrorLastCallFlag() )  
    ErrorObj obj = PopLastError();
```

Error Stack

Or query stack later:

```
while ( GetNumTotalErrors() > 0 )  
{  
    ErrorObj err = PopLastError();  
    Print( err.GetErrorString() );  
}
```

Results

- use to avoid parsing files
- database of named vectors of values (data)
- data from computation (i.e. mass props)
- data from geometry if requested (points, triangles...)

Useful functions:

GetAllResultsNames - returns all results names in database

FindResultsID - find id given name and index

GetAllDataNames - returns all data names given results id

Get<Type>Results - returns an array of data given results id, data name, index

WriteResultsCSVFile - writes everything in database in a file (comma sep)

Example - CreateEditGeom.as

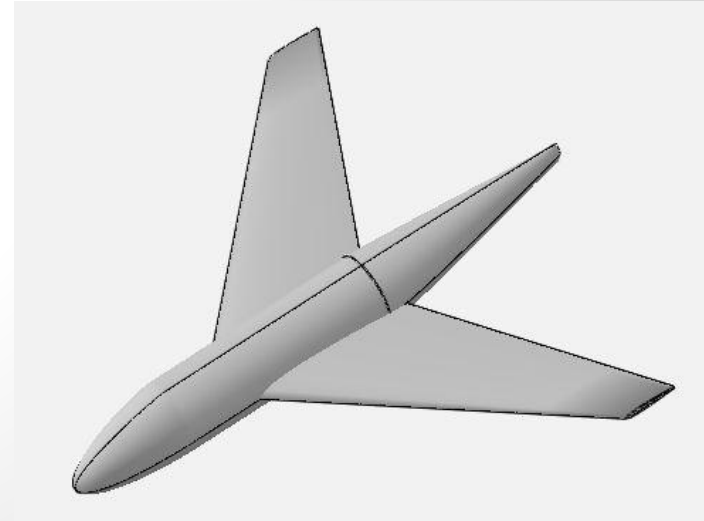
```
void main()
{
    string fid = AddGeom( "TransportFuse", "" );           // Add Fuselage
    string len_id = GetParm( fid, "Length", "Design" );
    SetParmVal( len_id, 30.0 );
    SetParmVal( GetParm( fid, "Diameter", "Design" ), 4.0 );

    string wid = AddGeom( "WING", "" );                   // Add Wing
    SetParmVal( GetParm( wid, "Span", "XSec_1" ), 15.0 );
    SetParmVal( GetParm( wid, "Root_Chord", "XSec_1" ), 10.0 );
    SetParmVal( GetParm( wid, "Tip_Chord", "XSec_1" ), 3.0 );
    SetParmVal( GetParm( wid, "Sweep", "XSec_1" ), 45.0 );
    SetParmVal( GetParm( wid, "X_Rel_Location", "XForm" ), 8.0 );
}
```

Example - CreateEditGeom.as

```
string xsec_surf = GetXSecSurf( wid, 0 );
while ( GetNumXSec(xsec_surf) > 2 )           // Remove Outboard Sections
{
    CutXSec( xsec_surf, GetNumXSec(xsec_surf)-1 );
}
Update();
WriteVSPFile( "TestWrite.vsp3", SET_ALL );    // Write To File

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



Example - RunCompGeom.as

```
void main()
{
    ReadVSPFile( "TestWrite.vsp3" );

    //==== Run Mass Properties ====//
    string sliced_mesh_id = ComputeCompGeom( SET_ALL, false, 0 );

    //==== Review Results Names ====//
    array<string> @res_names_array = GetAllResultsNames();
    for ( uint i = 0 ; i < res_names_array.size() ; i++ )
    {
        Print( "Results: ", false );
        Print( res_names_array[i] );
    }
}
```

```
Data: Num_Degen_Tris_Removed
Data: Num_Meshes
Data: Num_Open_Meshes_Merged
Data: Num_Open_Meshes_Removed
Data: Theo_Area
Data: Theo_Vol
Data: Total_Num_Meshes
Data: Total_Num_Tris
Data: Total_Theo_Area
Data: Total_Theo_Vol
Data: Total_Wet_Area
Data: Total_Wet_Vol
Data: Wet_Area
Data: Wet_Vol
Total Wet Area = 615.741235
```

Example - RunCompGeom.as

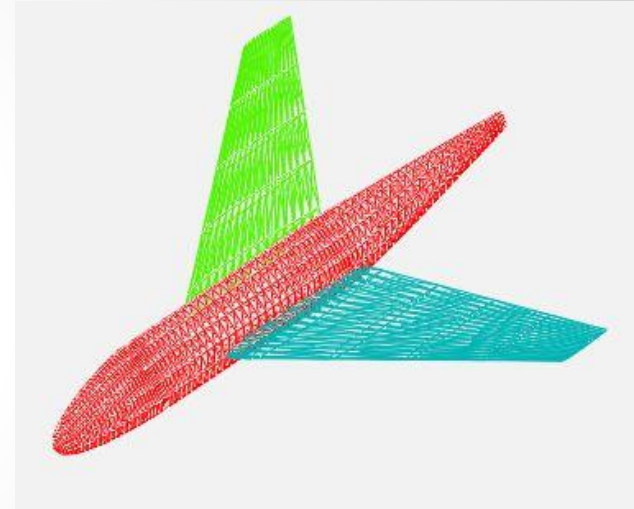
```
string res_id = FindResultsID(res_names_array[0] );           // Data Names  
array< string > @data_names_array = GetAllDataNames( res_id );
```

```
for ( uint i = 0 ; i < data_names_array.size() ; i++ )  
{  
    Print( "Data: ", false );  
    Print( data_names_array[i] );  
}
```

```
//==== Find Total Mass ====
```

```
array<double> @double_arr = GetDoubleResults( res_id, "Total_Wet_Area" );  
Print("Total Wet Area = ", false );  
Print( double_arr[0] );
```

```
}
```



Write and Run Your Own Script