

OpenCascade Draw Test Harness

eryar@163.com

Abstract. Draw is a command interpreter based on Tcl/Tk and a graphical system used to test and demonstrate OpenCascade modeling libraries. It provides a flexible and easy to use means of testing and demonstrating the OCCT modeling libraries. By a given command of Tcl, you can find the implementation code for that command by OpenCascade, the code could be as a demo to use of OCC modeling libraries. So this is the most effectively way of use OpenCascade.

Key words. OpenCascade, Tcl/Tk, Testing, Scripts

1. Introduction

Draw Test Harness 提供了灵活和易于使用的方式来测试、演示 OpenCascade 的建模算法。可用来交互的创建、显示和修改曲线、曲面及拓扑形状。你也可以自己写 Tcl 脚本来自定义 Draw 或使用脚本来实现自动化测试。也可以为 Draw 添加新的自定义的命令。

因为 Draw 是基于 Tcl/Tk 脚本的，所以可以根据 OpenCascade 自定义的命令名称来找到相关的实现，这个实现应该是使用 OpenCascade 最有效的方式，也是官方使用 OpenCascade 的方式，具有很好的参考价值。

还可在 Draw 中使用 Tcl 脚本来实现概念建模，如果结果和预期一致，可再把 Tcl 命令相对应的 C++代码写到程序中，提高编程效率。

本文主要介绍如何根据 OpenCascade 中实现的命令找到相关的实现代码，作为编程时的参考，避免了用 C++代码来实现一些功能时自己来摸索，从而浪费不必要的时间。

2. Using Tcl/Tk

先给出 Draw 中使用脚本出来的效果的样子，再介绍找出命令实现代码的方法。如下图所示为用脚本在 Draw 中产生的效果：

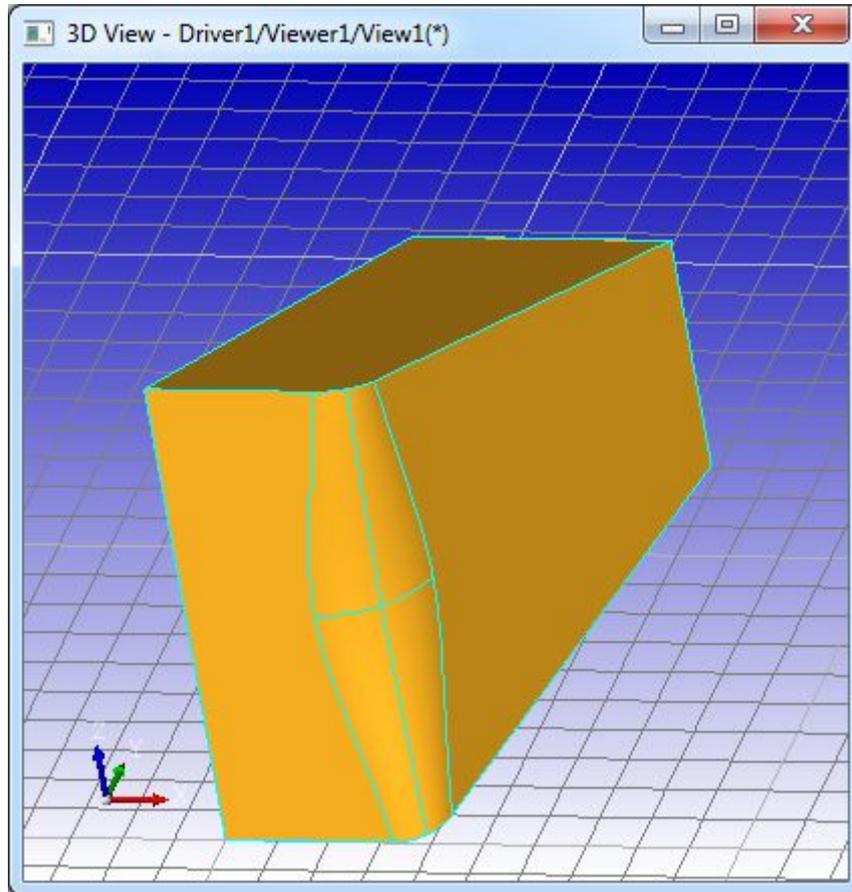


Figure 2.1 A blend test in Draw Test Harness

其实现的脚本如下所示：

```
pload ALL

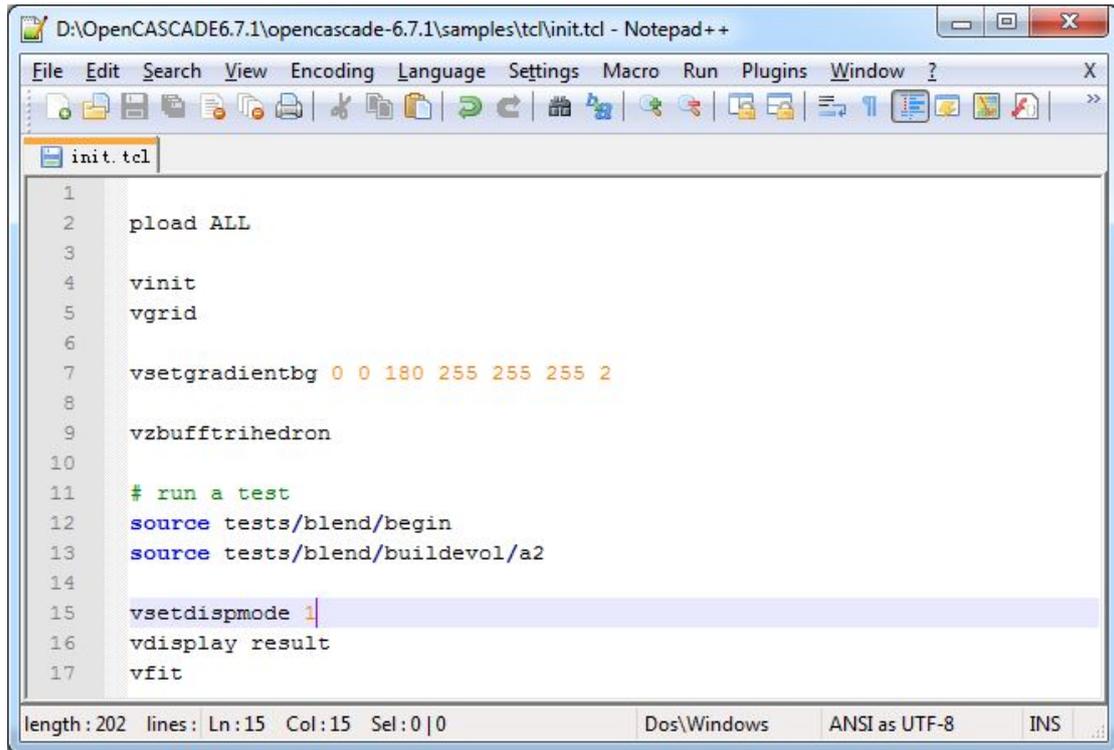
vinit
vgrid

vsetgradientbg 0 0 180 255 255 255 2

vzbufftrihedron

# run a test
source tests/blend/begin
source tests/blend/buildevol/a2

vsetdispmode 1
vdisplay result
vfit
```



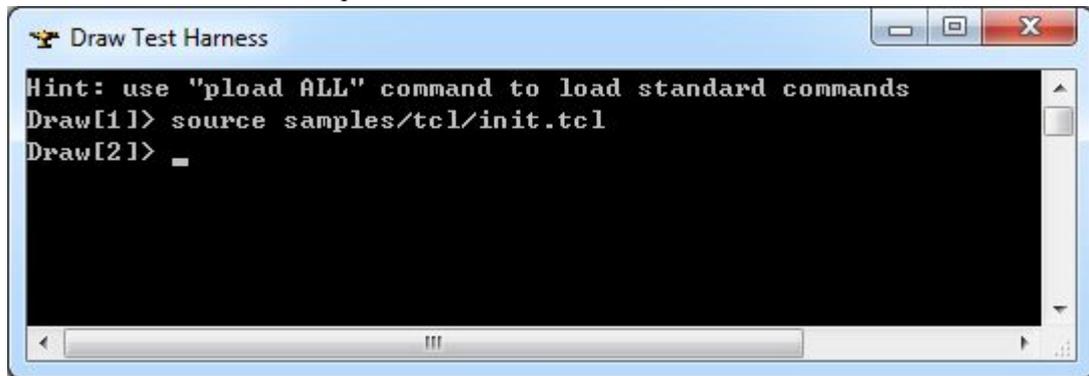
The image shows a Notepad++ window titled "D:\OpenCASCADE6.7.1\opencascade-6.7.1\samples\tcl\init.tcl - Notepad++". The window contains a Tcl script with the following lines:

```
1
2  pload ALL
3
4  vinit
5  vgrid
6
7  vsetgradientbg 0 0 180 255 255 255 2
8
9  vzbufftrihedron
10
11  # run a test
12  source tests/blend/begin
13  source tests/blend/buildevol/a2
14
15  vsetdispmode 1
16  vdisplay result
17  vfit
```

The status bar at the bottom indicates: length: 202 lines: Ln: 15 Col: 15 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS

Figure 2.2 Tcl script

将上述脚本代码保存到 samples/tcl 文件夹下，文件名为 init.tcl，再启动 Draw，输入命令：



The image shows a terminal window titled "Draw Test Harness". The terminal displays the following text:

```
Hint: use "pload ALL" command to load standard commands
Draw[1]> source samples/tcl/init.tcl
Draw[2]> _
```

Figure 2.3 Use the Tcl file

3. Find code from Draw Test

因为 Draw 是基于 Tcl/Tk 实现的解释器，所以根据 Tcl 中自定义命令的方法，可以找出 OpenCascade 中这些命令的实现代码。更方便的做法是直接输入命令名称，再在所有的源代码中搜索相关文件即可。如上述脚本中的 vgrid 命令，是打开、关闭栅格，可以直接搜索 vgrid，得到结果如下所示：

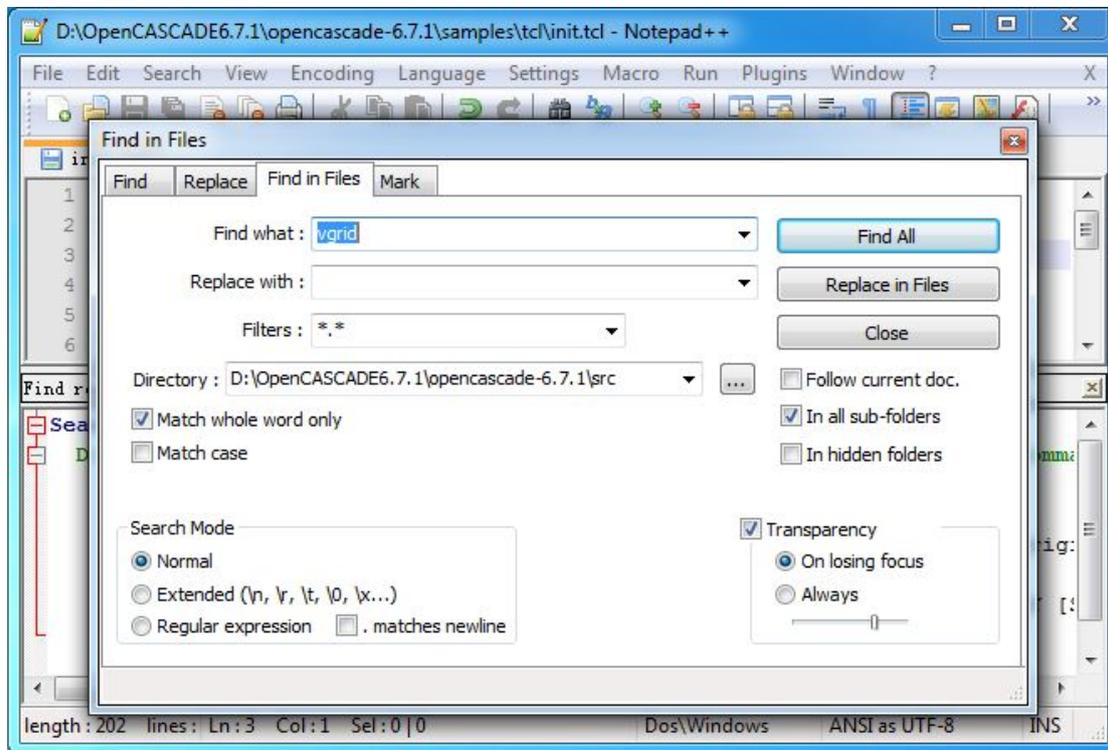


Figure 3.1 Find command in Files

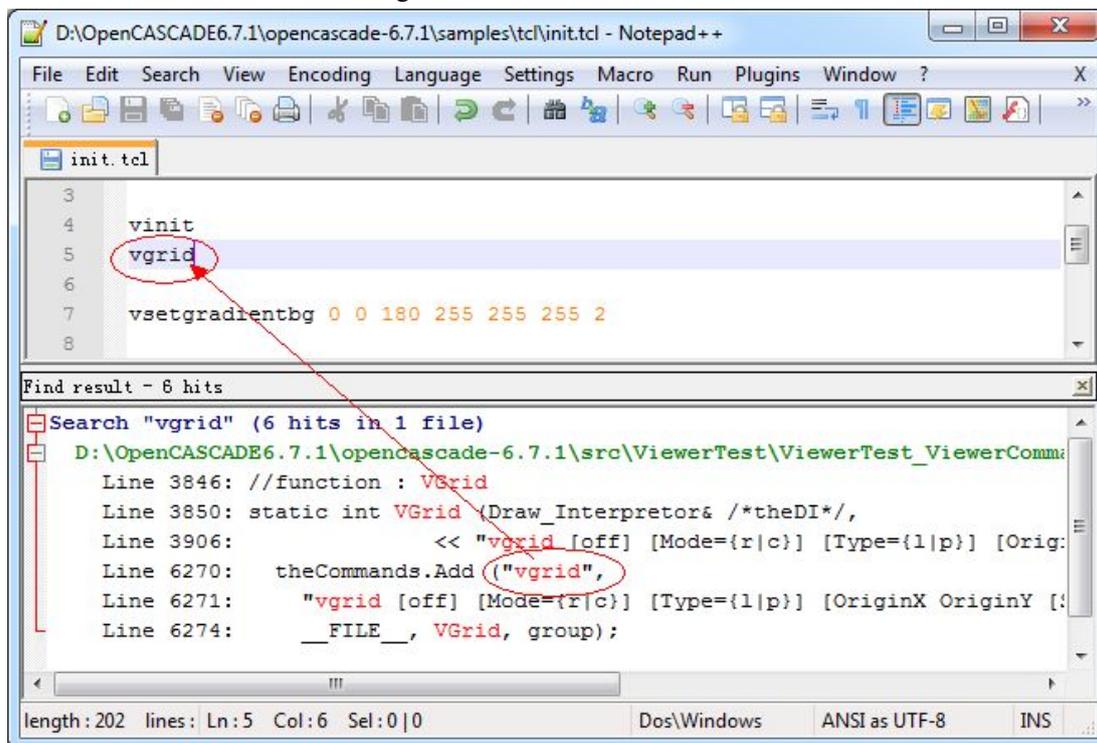


Figure 3.2 Find Result of the Command

得到相关的代码使用方法如下图所示：

```
3912 if (aType == Aspect_GT_Rectangular)
3913 {
3914     Quantity_Length aRStepX, aRStepY;
3915     aViewer->RectangularGridValues (anOriginX, anOriginY, aRStepX, aRStepY, aRotAngle);
3916
3917     anOriginX = Draw::Atof (theArgVec[anIter++]);
3918     anOriginY = Draw::Atof (theArgVec[anIter++]);
3919     if (aTail == 5)
3920     {
3921         aRStepX = Draw::Atof (theArgVec[anIter++]);
3922         aRStepY = Draw::Atof (theArgVec[anIter++]);
3923         aRotAngle = Draw::Atof (theArgVec[anIter++]);
3924     }
3925     aViewer->SetRectangularGridValues (anOriginX, anOriginY, aRStepX, aRStepY, aRotAngle);
3926     aViewer->ActivateGrid (aType, aMode);
3927 }
3928 else if (aType == Aspect_GT_Circular)
3929 {
3930     Quantity_Length aRadiusStep;
3931     Standard_Integer aDivisionNumber;
3932     aViewer->CircularGridValues (anOriginX, anOriginY, aRadiusStep, aDivisionNumber, aRotAngle);
3933
3934     anOriginX = Draw::Atof (theArgVec[anIter++]);
3935     anOriginY = Draw::Atof (theArgVec[anIter++]);
3936     if (aTail == 5)
3937     {
3938         aRadiusStep = Draw::Atof (theArgVec[anIter++]);
3939         aDivisionNumber = Draw::Atoi (theArgVec[anIter++]);
3940         aRotAngle = Draw::Atof (theArgVec[anIter++]);
3941     }
3942
3943     aViewer->SetCircularGridValues (anOriginX, anOriginY, aRadiusStep, aDivisionNumber, aRotAngle);
3944     aViewer->ActivateGrid (aType, aMode);
3945 }
```

Figure 3.3 The Command function code

依此类推，其他 Draw 中的命令都可以找到相关的实现代码，这些代码都可作为编程的依据，有助于 OpenCascade 的正确、高效使用。

4. Conclusion

从上可知，如果需要 OpenCascade 的什么功能，可以先在 Draw Test Harness 中试试看，如果在 Draw Test Harness 中可以实现，再找出其实现那个命令的代码，再参考代码实现所需要功能即可。这样编程使用 OpenCascade 应该是轻松和高效的，因为 Tcl 是解释执行，不像 C++那样需要编译时间。

5. References

1. Tcl and the Tk Toolkit
2. Practical Programming in Tcl and Tk
3. Tcl/Tk A Developer's Guide
4. OpenCascade Test Harness User's Guide
5. <http://sourceforge.net/projects/tcl/>
5. <http://www.tcl.tk/>