Open CASCADE Technology
7.7.0

Novice Guide

November 2, 2022

Contents

1   Modeling with OCCT: Key differences

Open CASCADE Technology (OCCT) is an object-oriented C++ framework designed for rapid production of sophisticated CAD/CAM/CAE applications. In other words, it provides endless possibilities for raw 2D and 3D modeling in C++ environment. Unlike end-user software, it is used by the application developers and therefore strongly differs from the most popular CAD/CAM/CAE software packages. OCCT provides building blocks enough for modeling, editing, visualization, and data interoperability of 2D and 3D objects.

By using OCCT, users can create the objects of their desire (or edit already existing ones) using raw code commands. It is a more complicated process than using GUI-based software, but it provides much more flexibility than even script-based manipulations that are available within existing CAD/CAM/CAE applications. However, to fully grasp the possibilities of OCCT it is best for the user to have previous experience in C++ at least at a basic level.

2 Understanding the principles

If you don't have any programming skills, grasping the full magnitude of OCCT workflow is still an option. The documentation for OCCT contains several entry points for new users. It will not explain all OCCT classes but will help to comprehend the workflow and help start thinking in terms of Open CASCADE Technology.

The most basic workflow is described in the OCCT Tutorial - this is an excellent starting point for new users. In this tutorial you will create a solid model step-by-step using different classes and methods. Each step of the tutorial contains code snippets and images.

The basics involved in the modeling process are explained. When the basics of OCCT are clear, the next logical step is to check out sample applications and examine those that suit your needs. For these, the best starting point is **OCCTOverview** located in /samples/qt subfolder of OCCT installation.

This sample provides code examples for several actions as well as visualization of these code snippets output. The Overview interface is dynamically changing based on selected **Category** at the menu. Additional menu buttons will appear, providing users with subcategories and relevant commands to select one of the actions. The result will appear in the viewer window, the code will appear at the top right, and in several cases the output will be produced at the bottom right window.
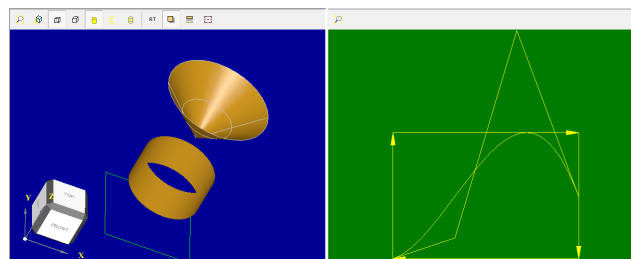

Figure 1: Comparison of 3D and 2D viewer windows

height=420px

The 3D viewer window has a row of preset interface buttons to customize the visual output.

Those buttons can be grouped into three types, left to right:

- View controls: **Fit all** and **Isometric**, will center the view and reset the camera angles respectively;

- Display mode customization: **HLR,** e.g. "Hidden line removal" (works only when shading is disabled) can be turned on and off; solid models may be displayed either in **Shading** or **Wireframe** modes. **Transparency** level may be set for models in shading mode;

- The last four buttons in a row are beautifiers enabling Ray-tracing engine and configuring it's parameters.

At the bottom left of the screen the orientation cube (trihedron) is located. The trihedron interactively shows the position of the camera in relation to the XYZ axis of the displayed data. The sides of the trihedron are labeled to help with orientation. Click on a side of the box to orient the camera view along the preferred axis.

The 2D viewer window lacks most of these elements and only have **Fit all** button.

The **Geometry** category of the Overview focuses on primitive objects like dots, lines (including vectors) or planes. These objects will appear in the viewer after the subcategory is selected. This section will demonstrate these entities both in 2D and 3D view mode and provide basic examples of parametric creation and data analysis.
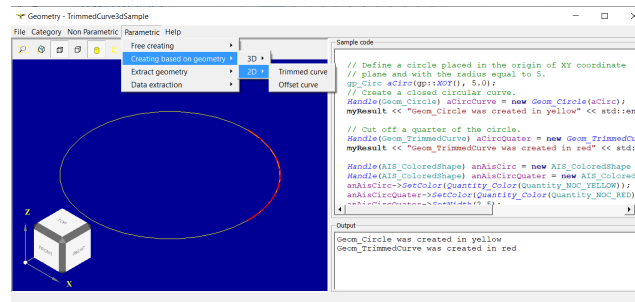
height=440px

The usage of the functions shown in the Overview is described more thoroughly at the Modeling data section of the documentation. Additionally, Modeling Algorithms are used in more complex cases.

The **Topology** section of the Overview demonstrates the functions used in 3D operations. Multiple use cases are provided, including different object intersections, modifying and calculations. Some of these use cases are described in the documentation, such as Inspector usage.
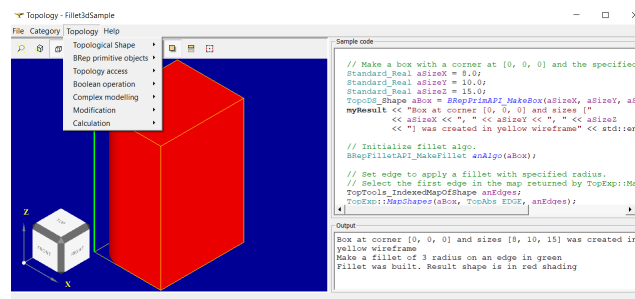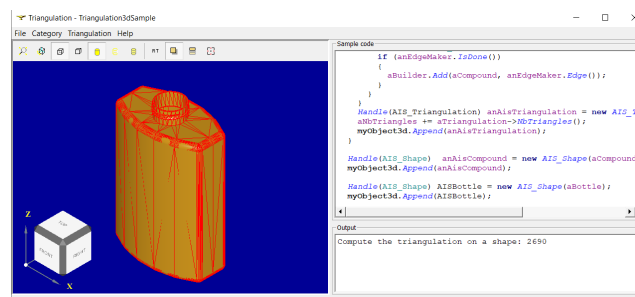

height=440px

The subsections are grouped as shown on the screenshot before. Most shapes and primitive objects are introduced and then followed by a set of operations and interactions.

The **Triangulation** segment allows computing the number of triangles on a shape.

This may be inspected via Poly_Triangulation Class Reference - a part of the Reference manual, an overall Open CASCADE code guide that may be used to inspect the key points in classes and their connections.
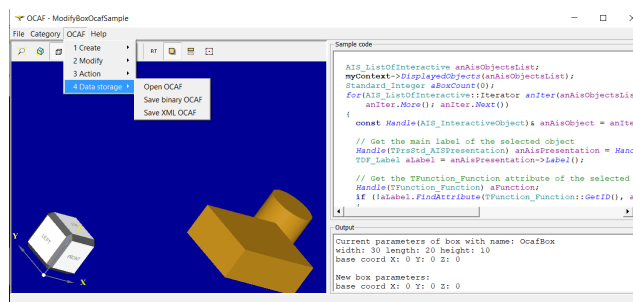

height=440px

The triangulation uses some of Mesh-related classes - see full description at Mesh documentation section.

The **Data exchange** section provides examples of how to export and import files of several different formats.
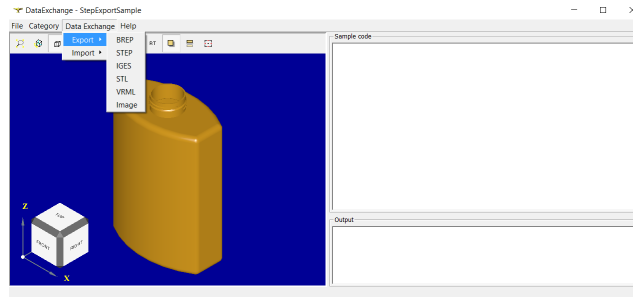
height=440px

The **OCAF** section gives an introduction for the Open CASCADE Application Framework functionality. To test these functions, create an object first (box or cylinder). After that, the object may be modified and saved. Actions are recorded and may be undone or redone.

height=440px

Viewers section demonstrates examples of the 2D and 3D visualization outputs. Check Visualization section of the documentation for a detailed description. In addition to these two samples, there are much more that might be of use to a new user based on their particular use case.

Check Readme files in the sample directories to learn more about samples compilation.

Note: source code for OCCTOverview is stored at 'samples/qt/OCCTOverview/src' folder in your OCCT root, and the source code files for examples presented in subsections are stored at 'samples/OCCTOverview/code folder'. Several utility classes that are not presented in the example window may be found in example source code files.

The overall classes introduction may be found in the Foundation Classes section of the documentation. The "↩ Introduction" section contains short descriptions of the most massive entries in the documentation.

3   Additional assistance

There are several places that may be of use for new users. The first one is `Training & E-learning` page that lists available trainings and describes their specifics.

The second one is the Overview documentation (this document is a part of it) - here you can find information that suits most of the use cases. This may seem overwhelming at first, but if you have the clear understanding of what do you seek, you will most likely find the required information.

Aside from the Overview documentation itself, the `Reference manual` is present. Use it to check classes descriptions, dependencies and examples. Additionally, there is a `Forum` where you can contact the OCCT community and developers.