Open CASCADE Technology
Version 7.5.0

Release Notes
Draft for beta version

October 7, 2020

Overview

Open CASCADE Technology 7.5.0 provides about 400 improvements and corrections over the previous release 7.4.0.

Open CASCADE Technology

Highlights

General

- Redesigned API of progress indicator for parallel tasks
- Support of compilation for WebAssembly (with Emscripten SDK)
- New printer Message_PrinterSystemLog for logging messages to system log

Modeling

- Support of progress indicator in BRepMesh
- New alternative algorithm for triangulation of 2d polygons
- Tool to remove internal sub-shapes (with INTERNAL orientation) from the shape keeping topological connectivity
- Allowed usage of multi-dimensional compound arguments for Boolean Cut and Common operations

Visualization

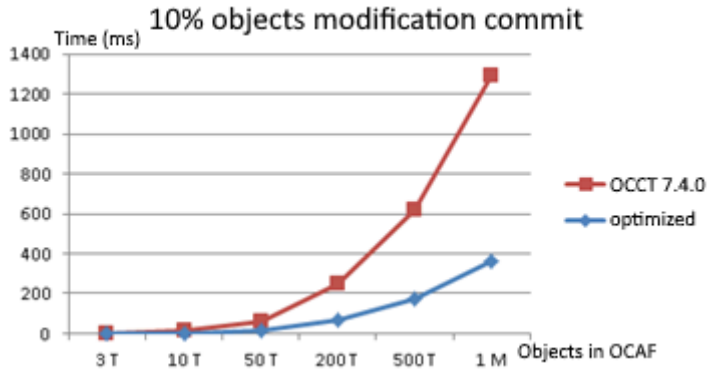
- Use of sRGB textures and render buffer
- PBR Metallic-roughness shading model
- Normal map texture support
- Option to compute BVH trees used for interactive selection in background thread
- Support of special-style font families and multi-font .ttc files in Font Manager

Data Exchange

- Support of reading STEP files containing non-Ascii symbols in text strings
- New API for STEP reader accepting C++ stream on input
- glTF 2.0 writer
- Improved performance of (ASCII) STL and OBJ readers

Application Framework

- Management of several documents (open, save, close, etc.) in parallel threads (one application per thread)
- Inheritance of attributes to reuse their persistence mechanisms
- Progress indicator in TDocStd_Application
- Optimization of Commit operation for large modifications



Draw Test Harness

- Colorized message output
- Support of Unicode symbols in DRAW console on Windows
- Flight-mode navigation in 3D viewer via WASD keys and 3D mouse input on Windows
- Experimental teleport-mode navigation in 3D viewer using OpenVR

Samples

- Unification of mouse gestures for 3D viewer manipulation in samples
- New WebGL viewer sample
- Update of JNI sample for Android Studio (from Eclipse project)
- New Qt OCCT Overview sample

Documentation

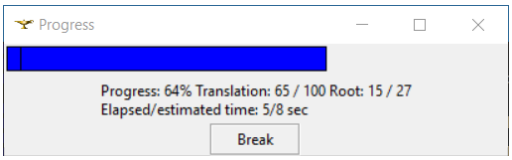
- Restructuring of OCCT documentation for easier orientation and higher user-friendliness

New Features

Thread-safe Progress Indicator

The progress indication mechanism has been revised to eliminate its weak points in previous design (leading to ambiguity and unprotected from an error-prone behavior).

New design allows using Progress Indicator in multithreaded algorithms in a more straightforward way with minimal overhead.



See documentation of the classes **Message_ProgressIndicator** and **Message_ProgressScope** for more details and examples.

Normal map texture support

OCCT 3D Viewer now supports additional texture maps: Normal map, Occlusion map and Emissive map. These texture maps can be used to artistically or realistically improve visual quality.

Normal texture may add extra details
like scratches or imprints without altering geometry.

sRGB textures and render buffer

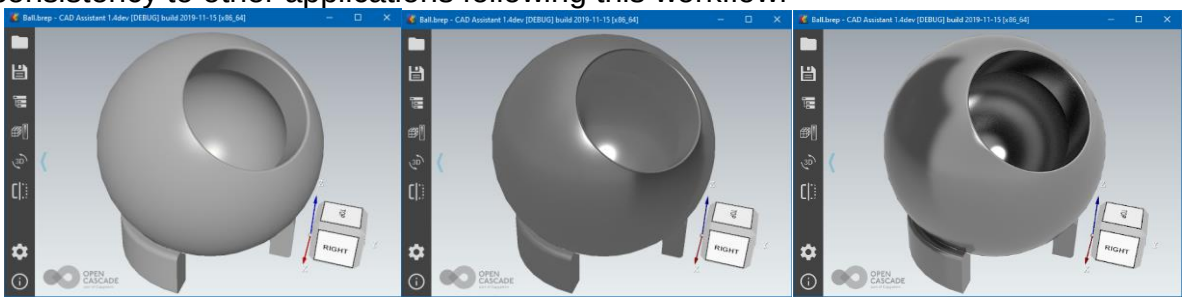
OCCT 3D Viewer has been improved to properly render an sRGB output (expected by default by modern displays).

This feature is essential for correct handling of semitransparent materials and for correct gradients (blended in linear color-space and not gamma-corrected one).

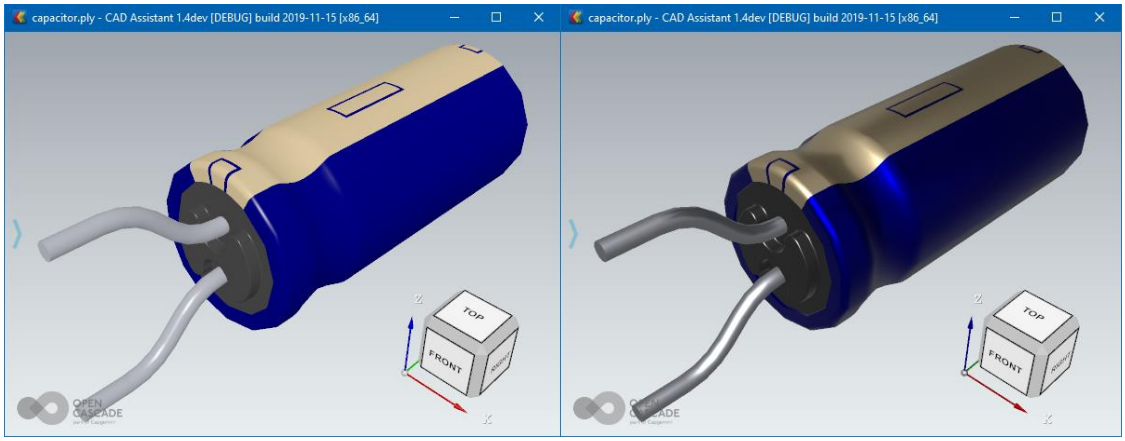
As a pre-requirement, **Quantity_Color** definition has been modified to expect RGB values in linear color space. New enumeration value **Quantity_TOC_sRGB** has been introduced for constructing Quantity_Color from sRGB values. Definition of standard OCCT materials and pre-defined named colors Quantity_NameOfColor has been updated accordingly.

PBR Metallic-Roughness material workflow

OCCT 3D Viewer now supports PBR (*Physically-Based Rendering*) Metallic-Roughness material workflow, which improves quality of metallic presentations and consistency to other applications following this workflow.



From left to right: Blinn-Phong shading, PBR Metallic-Roughness, Path Tracing.



Blinn-Phong shading (left) vs PBR Metallic-Roughness (right).

glTF 2.0 writer

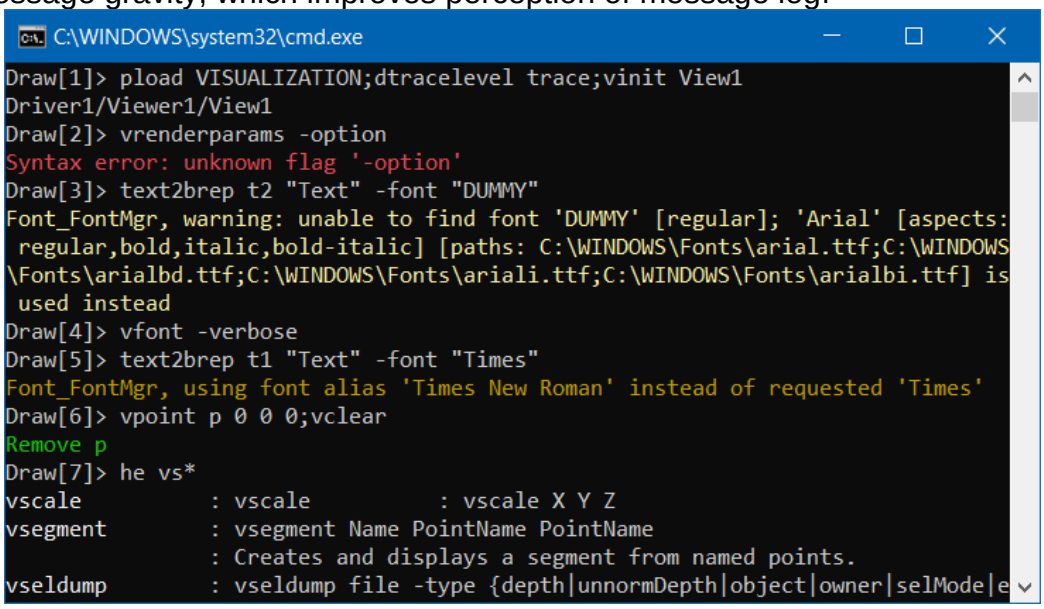
OCCT Data Exchange functionality has been extended with glTF 2.0 writer support provided by **RWGltf_CafWriter** class.

In Draw Harness conversion of STEP model into glTF file may look like this:

```
pload XDE OCAF VISUALIZATION

# read STEP file into XCAF document
ReadStep D [locate_data_file linkrods.step]

# auto-triangulate and display XCAF document
# in 3D Viewer with colors
vinit View1
XDisplay D -dispMode 1 -explore
vfit

# convert XCAF document into binary glTF file
WriteGltf D linkrods.glb
```

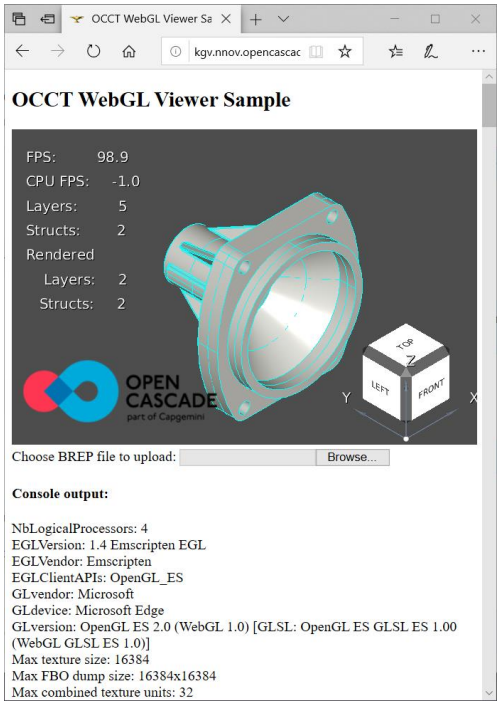

Draw Harness colored output into console

Message_PrinterOStream now supports colored output into the terminal depending on message gravity, which improves perception of message log.



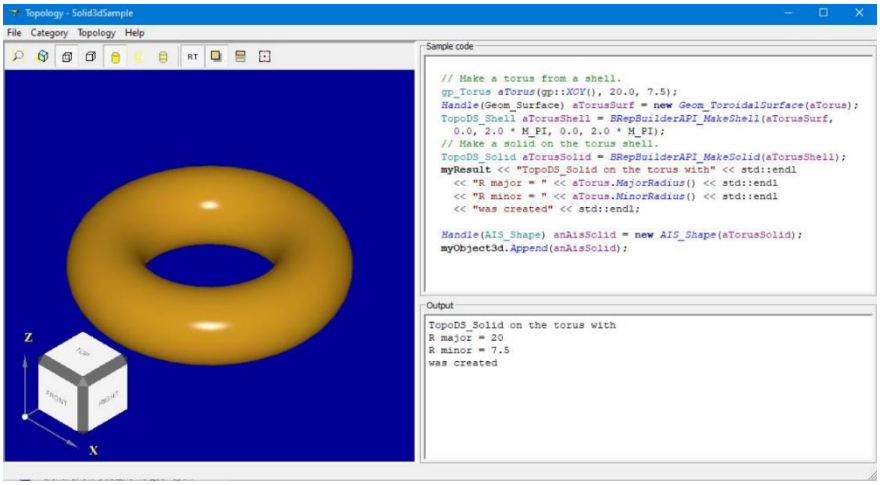
WebGL Viewer sample

New sample has been added to the repository demonstrating the usage of OCCT 3D Viewer built as WebAssembly within the browser.



Qt OCCT Overview sample

New sample has been added with GUI based on Qt framework and demonstrating use of basic Open CASCADE functionality via set of small code snippets.



Draw Harness VR teleport

Draw Harness 3D Viewer received several new features for navigating through large real-size models:

- Flight navigation via WASD and arrow keys (in perspective projection view).
- Flight navigation via 3D Mouse controller (Windows only).
- Teleport-style navigation via OpenVR controllers in VR view mode (requires OCCT being built with the **HAVE_OPENVR** flag).

To use this functionality, the loaded model in Draw Harness should be of room-scale or larger size and defined in proper length units, millimeters by default.

VR requires OpenVR drivers being installed on the computer - e.g. SteamVR with compatible VR headset. Draw Harness comes with HTC Vive controller bindings with more or less common layout for activating a teleportation via the top part of a navigation pad; trigger button can be used for highlighting (squeeze) and selecting (click) objects. Bindings for other controllers can be configured in the SteamVR configurator.

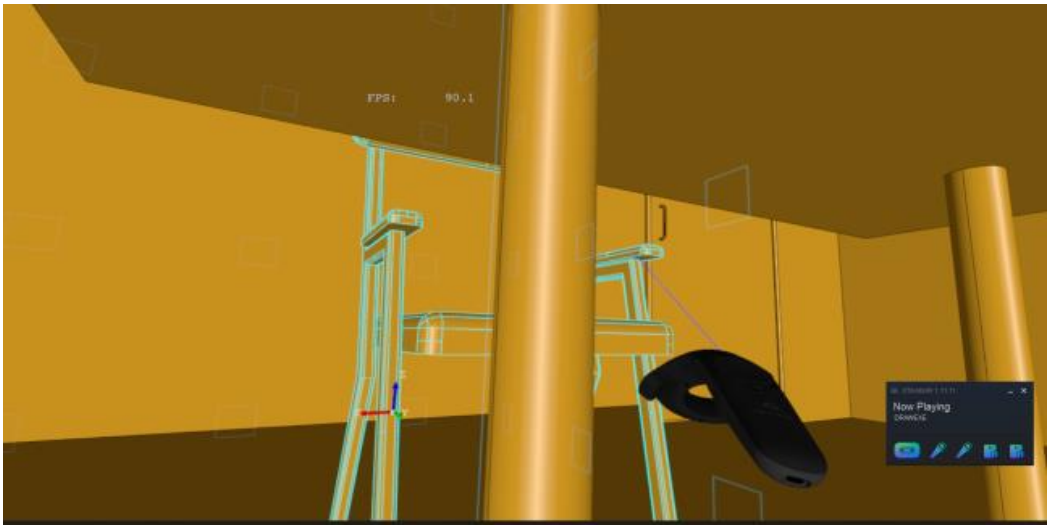
VR support is managed by Aspect_XRSession class, and can be enabled just like a common stereoscopic output with new value **Graphic3d_StereoMode_OpenVR** in case if application relies on **AIS_ViewController** for 3D Viewer input:

```
Handle(V3d_View) theView;
theView->ChangeRenderingParams().StereoMode = Graphic3d_StereoMode_OpenVR;
theView->Camera()->SetProjectionType (Graphic3d_Camera::Projection_Stereo);
```


Here is a sample Tcl script creating a scene of boxes:

```
dtracelevel trace
pload XDE MODELING VISUALIZATION OCAF
vinit View1 -width 768 -height 768
vzbufftrihedron
vrenderparams -perfCounters fps
for {set i 0} {$i < 5} {incr i 1} {
  for {set j 0} {$j < 5} {incr j 1} {
    for {set k 0} {$k < 5} {incr k 1} {
      box b$i$j$k $i $j $k 0.5 0.5 0.5
      vdisplay -noupdate -dispMode 1 b$i$j$k
    }
  }
}


vfit
vcamera -persp -lockZUp
vright
vrenderparams -msaa 4
vstereo -mode openvr -mirrorComposer 1 -unitFactor 1
vglinfo
```

Before entering into VR setup, it is necessary to alight up direction with expected one (like "*vcamera -persp -lockZUp*" for Z-up orientation), as VR will use current camera orientation as the reference for a room-scale navigation. In case of successful setup, "*vglinfo*" should print information about detected VR setup and VR headset should show 3D viewer content with mirrored output in a window. Put "*vstereo off*" to close VR session.



Documentation structure

Structure of OCCT Documentation has been revised to be more user-friendly:

- “Technical Overview” section content is merged into “Introduction” (former "Overview")
- Instructions for building OCCT building procedures are grouped by operating system and put into the dedicated section “Build, Debug and Upgrade”
- Description of samples and tutorial are moved into new section “Tutorials and Samples”
- “Developer Guides” section is removed (to avoid confusion between the User and Developer term), most of documents are put in the new section “Contribution”
- New section "Specifications" is added for documents providing technical details on implementation of different components of OCCT

Open CASCADE Technology

