

Open CASCADE Technology Maintenance release 5.1.3

Release Notes

Overview

Open CASCADE Technology Maintenance release 5.1.3 includes some bug fixes, new features and improvements over the previous maintenance release 5.1.2.

Note:

Due to API changes, this Version 5.1.3 is binary incompatible with the previous versions of Open CASCADE Technology, so applications linked against the latter must be recompiled to run with 5.1.3.

Highlights

- ▶ **TWO MORE OPEN CASCADE TECHNOLOGY ALGORITHMS (LOCAL OPERATION AND 3D OFFSETS) HAVE BEEN PORTED TO NEW BOOLEAN OPERATIONS THEREBY GAINING BETTER QUALITY AND RELIABILITY.**
- ▶ **TEXT OUTPUT CAPABILITIES OF OPEN CASCADE 3D VIEWER (WINDOWS PLATFORM ONLY) HAVE BEEN EXTENDED WITH THE POSSIBILITY TO USE TEXTURE-MAPPED FONTS AS AN ALTERNATIVE TO TRADITIONAL BITMAP FONTS.**
- ▶ **A PATCH HAS BEEN APPLIED ALLOWING TO READ INFORMATION ABOUT MATERIALS AND GEOMETRIC DIMENSIONS AND TOLERANCES FROM STEP FILES USING THE XDE READER.**

Table of Contents

- [**New Features**](#)
 - ⇒ [Data Exchange](#)
 - ⇒ [Visualization](#)
- [**Improvements**](#)
 - ⇒ [Modeling Algorithms](#)
 - ⇒ [Application Framework](#)
 - ⇒ [Visualization](#)
- [**Bug Fixes**](#)



New Features

Data Exchange

- In the framework of regular CAX-IF activity, Open CASCADE has extended its STEP Interface to handle STEP materials, dimensions and geometric tolerances.

Visualization

- Texture mapped fonts are now available on Windows platform as an alternative and more efficient method for text output in Open CASCADE 3D viewer.

[See below for more information](#)

Improvements

Modeling Algorithms

- The Boolean operations algorithm has been corrected to process the computation of a 2d curve corresponding to a 3d curve at the intersection of faces. No exception is raised now during the cut operation.
- Some limitations of New Boolean operations concerning the impossibility to process shapes with large tolerances of edges and vertices are solved, including cases where the processed shapes have same domain faces and same domain edges. Additional topological verification to perform (if possible) an immediate correction of building result has been implemented into the New Boolean operations algorithm.
- Projection of a 3d point on a surface of revolution is now computed correctly for the case when the axis of revolution does not lie in the plane of the revolved periodic curve.
- The shape analyzer now processes tiny faces more correctly than before.
- The problem in the fillet algorithm caused by the variation of input data precision influencing the result of calculation has been fixed.
- The BRepOffsetAPI_MakePipeShell algorithm can now process pipes in the sweep mode set either to IsGuideACwithContact or to IsGuidePlanwithContact.
- Now the algorithm of BRepOffsetAPI_MakePipeShell correctly processes the cases with linear parts of spines and profiles.
- The definition of parameter on a path corresponding to a section has been corrected, so the resulting swept surface is now built correctly.





 **OCAF**

- A possibility has been added to remove drivers from the static maps `TPrsStd_DriverTable` and `TFunction_DriverTable` with the help of methods:
 - `RemoveDriver` - removes a driver with the given GUID,
 - `Clear` - removes all drivers.

This improvement allows to :

- Avoid problems connected to the unloading order of modules where the code of custom drivers is defined. Previously an exception could be raised if a custom module had been unloaded before the destruction of tables.
- Change presentation and function drivers at any point in the program thus implementing some sophisticated behavior of an application.

Note:

If the `TPrsStd_DriverTable::Clear` method is used, the `InitStandardDrivers` should be called to fill the table with standard drivers.

 **Visualization**

- Text output capabilities of Open CASCADE 3D viewer (Windows platform only) have been extended with the possibility to use texture mapped fonts as an alternative to traditional bitmap fonts.

This new feature makes text output on Windows several times faster as compared to the old approach, provided that graphic hardware and video drivers support hardware acceleration of texture operations. This is especially important for applications dealing with a large number of text labels displayed in the 3D viewer, since a 3D scene is updated very slowly with several thousand text labels displayed with the help of bitmap fonts.

For easy control over the new functionality, `CSF_TEX_FONT` environment variable can be used. Several values can be assigned to it having different meaning as described below:

less or equal to zero	texture mapped fonts cannot be used; only bitmap fonts are available;
1	texture mapped fonts are enabled, bitmap fonts are used by default (old text output method);
2	texture mapped fonts are enabled, texture mapped fonts are used by default (new method).

Hence to use the new text output method in an application, it is enough to set the `CSF_TEX_FONT` environment variable to 2, and no source code modification is needed. If the variable is not defined then use of texture mapped fonts is disabled and old approach is used.

To switch texture mapped fonts on / off in run-time, `CSF_TEX_FONT` should be set to 1 or 2, and `Graphic3d_AspectText3d::SetTextureMappedFont(Standard_Boolean)` method should be used with the `Standard_True / Standard_False` argument. First, a `Graphic3d_AspectText3d` object should be prepared using the above-described



method, and then this object can be used in a usual manner, i.e. passed to the `Graphic3d_Group::SetPrimitivesAspect()` method. Without explicit calls to `Graphic3d_AspectText3d::SetTexturedMappedFont()` the objects get created with the value defined by the `CSF_TEX_FONT` variable (see the table above).

Note: This use convention (based on system variable) is temporary and will be changed in the next version of Open CASCADE Technology.

- Interface of `AIS_InteractiveObject`, `AIS_InteractiveContext` and `Graphic3d_AspectFillArea3d` classes has been extended to allow drawing of partially or fully overlapping polygons in the shading mode without any visual artifacts with the help of Open CASCADE Technology 3D graphic capabilities. Additionally, this improvement allows to avoid visual artifacts when drawing a line lying in plane with some shaded polygon.

Usually, when you display two overlapping or touching surfaces or meshes in the shading mode, it results in "blinking" in coincident areas during 3D view rotation or zooming. This is default OpenGL behavior (the depth buffer simply cannot resolve which pixel of the two should be displayed).

Fortunately, OpenGL provides developers with a polygon offset mechanism that allows to avoid this undesirable effect. There are three parameters controlling polygon offsets:

- `mode`: it is possible to turn polygon offsets on and off for certain polygon visualization methods (filled, lines, points) independently;
- `factor`: when non-zero, this parameter adds more offset for polygons having a larger angle with respect to the viewport;
- `units`: this value is added to the viewport Z co-ordinates of polygon's pixels;

when negative, it makes pixels closer to the viewport, and otherwise moves pixels away from the viewport.

From now on, in Open CASCADE Technology there are two ways to deal with the problem:

1. `AIS_InteractiveObject::SetPolygonOffsets(mode, factor, units)` method.

You can move one of the two 3D objects (they should inherit the `AIS_InteractiveObject` class) a bit away from the eye by turning on polygon offsets for this object:

```
Handle(AIS_Shape) aShape = getAISShape(); // somehow
obtain your interactive object

Handle(AIS_InteractiveContext) aCTX = getAISContext(); //
get interactive context

aShape->SetPolygonOffsets(Aspect_POM_Fill, 1., 1.); // set
up polygon offsets for filled polygons
aCTX->Display(aShape);
```

This sample code makes the OpenGL depth buffer consider `aShape` presentation as being at a slightly longer distance from the eye than it actually is.

For the second object, polygon offsets can be turned off explicitly:



```
aSecondShape->SetPolygonOffsets(Aspect_POM_Off);
```

The result of this is aSecondShape presentation being visible in the areas of overlapping (and aShape being invisible in such areas).

2. Graphic3d_AspectFillArea3d::SetPolygonOffsets(mode, factor, units) method.

A more flexible and sophisticated way is to control polygon offsets at the level of Graphic3d_Group objects, which are created by the Compute() method of your interactive object. This approach allows to display overlapping primitives belonging to the same object presentation. If you choose this way, you should turn polygon offsets on for Grpaphic3d_Group objects which should have a "priority" in your 3D scene, and for all the others polygon offsets should be off:

```
Prs3d_Root::NewGroup(aPrs3d);
  Handle(Graphic3d_Group)          aGrp          =
Prs3d_Root::CurrentGroup(aPrs3d);
  Handle(Graphic3d_AspectFillArea3d) aFillAsp =
myDrawer->ShadingAspect()->Aspect();
  aFillAsp->SetPolygonOffsets(Aspect_POM_All, 1., 1.);
  aGrp->SetPrimitivesAspect(aFillAsp);
  // here you add graphic primitives to aGrp
  ...
Prs3d_Root::NewGroup(aPrs3d);
  Handle(Graphic3d_Group)          aPriorGrp     =
Prs3d_Root::CurrentGroup(aPrs3d);
  aFillAsp->SetPolygonOffsets(Aspect_POM_Off);
  aPriorGrp->SetPrimitivesAspect(aFillAsp);
  // here you add graphic primitives to aPriorGrp
```

NOTE:

1. Polygon offsets have an impact on polygons only (triangles, quadrangles, etc.), irrespective of the mode they are drawn in (shading, wireframe or points). Such primitives as vertices and lines are not affected by this mechanism.
2. By default, all object presentations have the following polygon offset parameters: mode = Aspect_POM_Fill, factor = 1.0, units = 0.0.

All Graphic3d_Groups have the polygon offset mode Aspect_POM_None by default, so as not to modify polygon offset settings for the corresponding interactive object.



Changes

Modeling Algorithms

- A new static function `IsSweepParallelSpine` has been added to `GeomFill_Sweep`. It checks whether a swept surface is parallel to the spine or not, so the canonic part is defined.

Bug Fixes



- Open CASCADE 5.1.3 incorporates **35** modifications (bug fixes, enhancements and other corrections) over the previous maintenance version 5.1.2. For details, refer to [Appendix 1](#).

Appendix 1: Open CASCADE 5.1.3 Bug Fixes

- [Data Exchange](#)
- [Foundation Classes](#)
- [Modeling Algorithms](#)
- [Application Framework](#)
- [Visualization](#)
- [Test Harness Draw](#)

Data Exchange, 8 bug fixes	
ID	Short Description
2513	Class MoniShapeOperAdv belongs to two toolkits
3997	No check of pointer is made for the null value
5130	Can not read STEP file with curves bounded by points
5187	XCAFDoc_DocumentTool::IsXCADFDocument is declared but not implemented
5192	After translation the resulting curves are incorrect in a STEP file.
5237	Progress Indicator inserted in the XSControl_TransferRoots
5303	Integration of changing STEP translator after CAX-IF TR13J
DXF572	Written DXF file can not be read by Volo View Express.
Foundation Classes, 1 bug fix	
ID	Short Description
5600	Corrupted header file NCollection_Stack.hxx
Modeling Algorithms, 19 bug fixes	
ID	Short Description
497	Results of cut and fuse operations are faulty shapes although arguments are valid.
578	Bug in BRepClass3d_SolidClassifier
943	Wrong projection of 3dcurve on surface.
3896	BRepCheck_Analyzer does not detect faulty faces in a solid shape.
4717	A regression of new Boolean Operations
4900	Incorrect work of DRAW command bfuse
5073	Problem with sweep
5093	Wrong result of the pipe algorithm
5110	Cut operation throws an exception
5112	Cut operation produces invalid shape
5113	Cut and common operations produce invalid results
5246	Bug of new Boolean Operations.
5335	Error in MakePipeShell: program hangs up
5336	Incorrect behavior of MakePipeShell algorithm
5579	Providing correct behavior of exception catching code both STL and non-STL streams
5581	Impossible to make an offset on a shape
5607	Porting of Local Operation algorithms to use New Boolean operations
5634	Porting of 3d Offset and Features algorithms to use New Boolean operations
5645	Useless messages in DRAW while performing command "mkoffset"
Application Framework, 1 bug fix	
ID	Short Description
5559	It is impossible to explicitly clear driver tables





Visualization, 4 bug fixes	
ID	Short Description
2934	Optimized performance on 3D text output in 3D viewer.
4692	It is impossible to highlight Graphic3d_ArrayOfPolylines primitives.
4844	Function V3d_View::Rotate uses static variables
4895	Eliminate Z buffer artifacts when some overlapped shaded polygons are drawn
Test Harness – Draw, 2 bug fixes	
ID	Short Description
3565	Dynamic loading of Draw Commands
5601	Multiple run of pload command leads to memory problems