
# Open CASCADE Technology and Products ver. 7.1.0 Minor Release

## Release Notes

## Overview

**Open CASCADE Technology and Products version 7.1.0** is a minor release, which includes more than **500** new features, improvements and bug fixes over major release 7.0.0.

## Highlights

**General**

- Support of Windows Store applications
- Definition of `Standard_Boolean` as `bool` instead of `unsigned int`
- Execution without need of setting environment variables

**Application framework**

- OCAF persistence without dynamic plugins
- Support of several attributes of one type at the same label

**Modeling algorithms**

- Improved 3D Offset operation in mode Complete with Joint type Intersection
- Calculation of the optimal axis-aligned bounding box for a shape
- Improved reliability of HLR algorithm
- Prevention of modification of original shapes in Modeling algorithms

**Visualization**

- Use of programmable pipeline (GLSL) by default on all platforms
- Improved rendering performance of Wireframe presentation
- Improved Shaded highlighting in AIS_Shape
- Improved clipping planes - capping preserving object material and better performance
- Support of perspective projection and selection for transformation persistent objects
- New presentation `AIS_Manipulator` for interactive object transformation
- New property in `AIS_Dimension` to set custom text value
- Type of angle and type of arrows properties in `AIS_AngleDimension`
- New optimized Path Tracing algorithm (adaptive screen space sampling)

**Data exchange**

- Improved support of `STEP AP242`, including PMI data, dimensions, and annotations

# Table of Contents


**New features** **4**
- *Support of Windows Store applications* 4
- *Execution without environment variables* 4
- *3D offset in mode Complete with Join Type Intersection* 6
- *Safe arguments processing by Boolean operations running with fuzzy option* 7
- *Optimal bounding box for a shape* 7
- *Prevent modification of original shapes in Modeling algorithms* 7
- *PipeShell algorithm provides history of generations* 7
- *Use of programmable OpenGL pipeline* 8
- *Highlighting style* 9
- *Ray-tracing renderer* 9
- *Clipping planes* 10
- *Manipulator object* 11
- *Support of several attributes of one type at the same label* 11
- *OCAF persistence without plugins* 12

**Modifications** **13**
- *Foundation Classes* **13**
- *Application Framework* **16**
- *Modeling Data* **18**
- *Modeling Algorithms* **19**
- *Mesh* **31**
- *Shape Healing* **32**
- *Visualization* **34**
- *VIS* **51**
- *Data Exchange* **52**
- *Draw* **54**
- *Samples* **55**
- *Configuration* **56**
- *Coding* **59**
- *Documentation* **61**
- *Added-value components* **62**
  - *ACIS-SAT Import/Export* 62
  - *Parasolid Import* 62
  - *DXF Import / Export* 63
  - *Canonical Recognition* 63
  - *Surfaces from Scattered Points* 63
  - *Mesh Framework* 64
  - *Express Mesh* 65
  - *Advanced Samples & Tools* 65
  - *Geodesic* 66

**Supported Platforms and Pre-requisites** **67**

# New features

## Support of Windows Store applications

On Windows, OCCT can now be built for Universal Windows Platform (UWP), and thus can be used for development of the applications intended for Windows Store, including those working on mobile devices running Windows 10 mobile (or other platforms supporting UWP).

When OCCT is built for UWP, some features are not available or have limited functionality:

- No system fonts are available in `Font_FontMgr` class;
- CPU and elapsed times are not available; classes `OSD_Timer` and `OSD_Chronometer` report zeros for all measurements;
- Environment variables can be used only if defined via `OSD_Environment` class;
- File permissions are not supported (ignored) by `OSD` classes;
- Information on process memory, attributes, user name, hostname etc. is not available; relevant classes from `OSD` package return zeros or empty strings;
- C signal handlers are not supported (`OSD::SetSignal()` does nothing);
- Visualization requires use of third-party OpenGL library, such as ANGLE;
- DRAW is not available.

To be able build for UWP, you need to have:

- Visual Studio 2015 with installed Universal Windows App Development Tools (Windows 10 SDK),
- CMake version 3.6.0 or newer,
- Windows 10 as host system,
- Freetype and ANGLE libraries built for UWP (WindowsStore), if you need to build Visualization module.

On configuration step of CMake, select "Visual Studio 14 2015" as generator and specify optional toolset, pointing to the file `adm/templates/uwp.toolchain.config.cmake` located within OCCT sources. Alternatively, run CMake with options `-DCMAKE_SYSTEM_NAME=WindowsStore -DCMAKE_SYSTEM_VERSION=10.0`.

Find more information in OCCT documentation and samples.

## Execution without environment variables

This release improves resource management in OCCT. Most mandatory resource files (including shaders for 3D viewer, unit definitions, and messages) are now embedded into the source code, so applications do not need to copy relevant resources along with OCCT binaries and set corresponding environment variables for proper execution.

This is especially helpful for platforms with limited file system permissions like iOS and Android, where installation of resource files is complicated. For desktop platforms, this makes OCCT installation more straightforward and allows avoiding possible collisions of resources between multiple versions.

The following resource files are concerned. (They still form a part of OCCT sources):

```
Resources
├─ Shaders
│   ├─ Display.fs
│   ├─ PathtraceBase.fs
│   ├─ RaytraceBase.fs
│   ├─ RaytraceRender.fs
│   ├─ RaytraceSmooth.fs
│   ├─ Declarations.glsl
│   ├─ DeclarationsImpl.glsl
│   └─ RaytraceBase.vs
├─ SHMessage
│   └─ SHAPE.us
├─ StdResource
│   ├─ Plugin
│   ├─ Standard
│   ├─ StandardLite
│   ├─ TObj
│   └─ XCAF
├─ TObj
│   └─ TObj.msg
├─ UnitsAPI
│   ├─ Lexi_Expr.dat
│   └─ Units.dat
└─ XSMessage
    └─ XSTEP.us
```

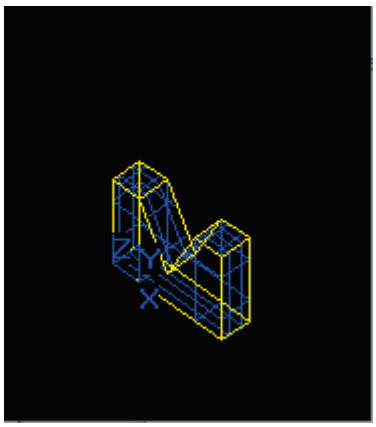
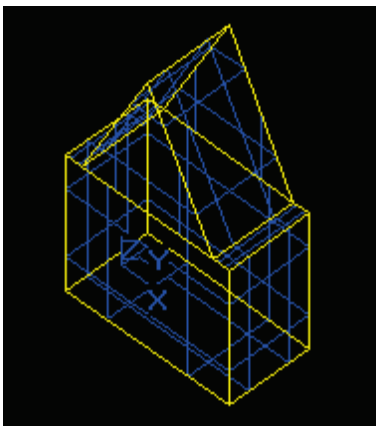
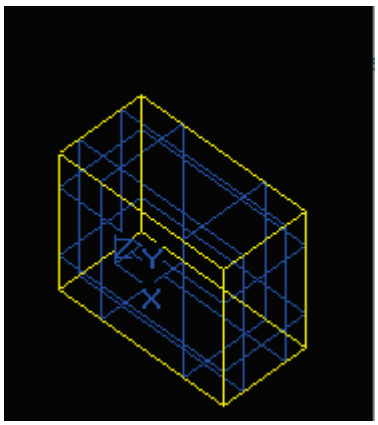
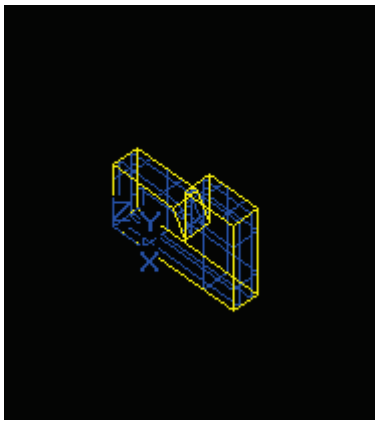
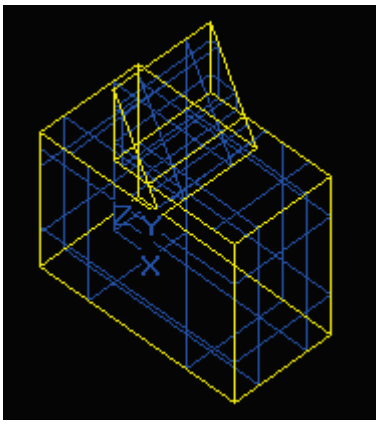
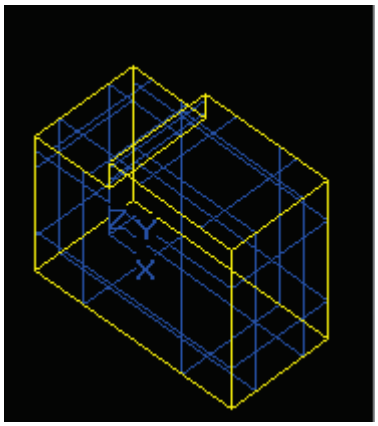
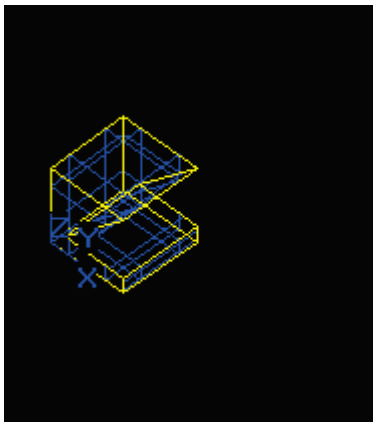
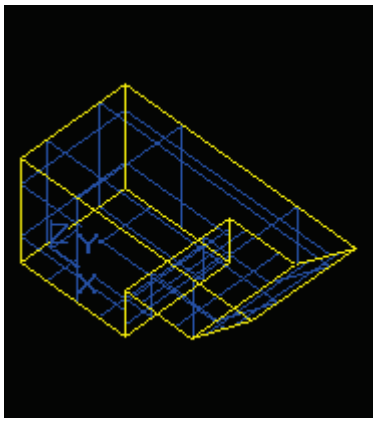
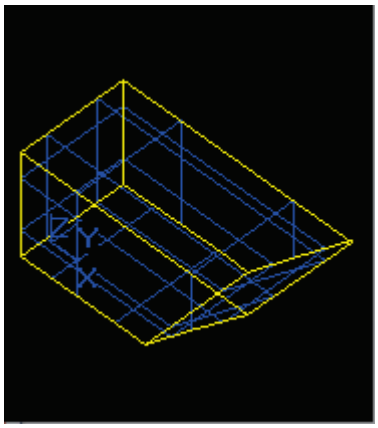
Note that resource files defining plugins for OCAF persistence (Plugin, etc.) are not embedded. To avoid the need of using environment variables and file resources for persistence, it should be loaded directly in C++ code.

## 3D offset in mode Complete with Join Type Intersection

3D Offset algorithm has been significantly improved to handle complex cases when working in mode "Complete" with Join type "Intersection". This improvement is limited to solids built with planar faces only.

This mode allows modifying the shape's topology by removing the colliding parts. In the previous implementation the result of the operation was built from the outermost offset faces, without checking the validity of the result. The current implementation is able to treat complex topological cases and avoid creation of invalid and unwanted parts such as "spikes" or inverted faces in result of the operation.
The examples below show the improvement of the algorithm in the most notable cases:

| Initial shape | OCCT 7.0 result | OCCT 7.1 result |
|---|---|---|
|  |  |  |
|  |  |  |
|  |  |  |

## Safe arguments processing by Boolean operations running with fuzzy option

In OCCT 7.1.0, Boolean operations with fuzzy option are protected from increase of the tolerance of the input shapes.

The mode of protection of input shapes from modification (turned on by call to `SetNonDestructive`) now supports fuzzy option as well. In the previous version the usage of fuzzy option in this mode was not supported and might lead to wrong results.

Thus, now fuzzy Boolean operations can be safely used in parametric modeling with naming.

## Optimal bounding box for a shape

OCCT 7.1.0 brings new methods for exact calculation of optimal axis-aligned bounding boxes of geometric entities. The old method of calculation of bounding boxes sometimes gives a very large box for B-spline and Bezier curves and surfaces. This is because it does not solve any optimization tasks (for the sake of speed). The new method works hard to compute the optimal bounding box, therefore sometimes calculation time can be greater than with the old method, but the result is accurate.

The API method for the new functionality for shapes is the static method `BRepBndLib::AddOptimal()`. Package `BndLib` contains the corresponding methods for curves and surfaces.

## Prevent modification of original shapes in Modeling algorithms

Many modeling algorithms use the class `BRepTools_Modifier`. One of its responsibilities is to ensure validity of output shape in terms of correct tolerance values of edges and vertices. In the previous versions increase of the tolerance of a vertex required for validity of the resulting shape could lead to modification of the input shape.

In OCCT 7.1.0, when it is needed to increase the tolerance of a vertex belonging to the original shape, the modifier makes a copy of that vertex in the output shape. Thus, now the algorithm ensures that the sub-shapes from the original shape are not changed during creation of the output shape.
For this, new vertexes, curves and surfaces are created before calling `Rebuild()` method and the copies of all vertexes are impacted by curve or surface modifications. Thus it is not necessary to create new vertices during recursive rebuilding of the entire shape.

This change concerns the algorithms located in the packages `BRepBuilderAPI`, `BRepFeat`, `BRepOffsetAPI_DraftAngle`, and some algorithms from Shape Healing library.

Compatibility with the old behavior of `BRepTools_Modifier` has been retained as an option. For that the new flag `MutableInput` has been added in the interface of the class. This flag can be used if it is needed to prevent copying of vertices for increase of their tolerances.

## PipeShell algorithm provides history of generations

Before OCCT 7.1.0 the method `BRepOffsetAPI_MakePipeShell::Generated()` returned empty list for most sub-shapes of the profile (section) shape. Due to this fact it was impossible to use this algorithm in `Naming`.

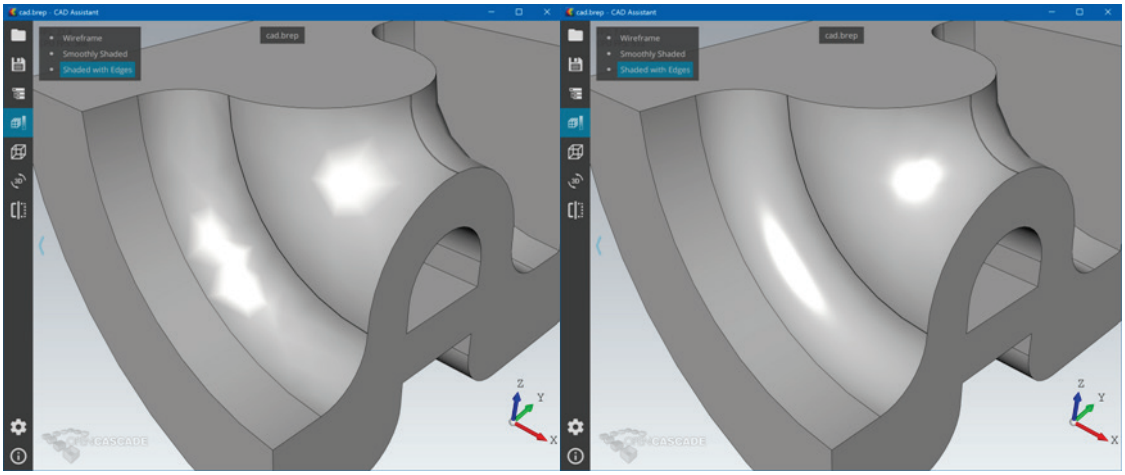
In OCCT 7.1.0, this method returns a list of shapes generated from sub-edges and sub-vertices of sections.

## Use of programmable OpenGL pipeline

OCCT 7.1.0 does not rely on the obsolete functionality provided by legacy versions of OpenGL standard, and now uses GLSL programs by default. Legacy rendering pipeline is still available in OCCT, but not maintained, and will be removed in the future.

Using programmable pipeline improves stability of 3D Viewer on modern graphic drivers, and opens a door to new visualization features, which were impossible before. E.g., the application can now enable Phong shading model instead of Gouraud for improved shading on low-poly models using `V3d_View::SetShadingModel (V3d_PHONG)`.



Gouraud (left) vs. Phong (right) shading on the same model

Programmable rendering pipeline (GLSL programs) has become available in OCCT several releases before, but until version 7.1.0 it did not provide all visualization features available within deprecated OpenGL features.

Within OCCT 7.1.0, all major visualization features are now available within programmable pipeline, and it is enabled by default. Next releases can introduce more new features that will be available only within programmable pipeline.

Note that flat shading model (V3d_FLAT) is not available in GLSL mode.

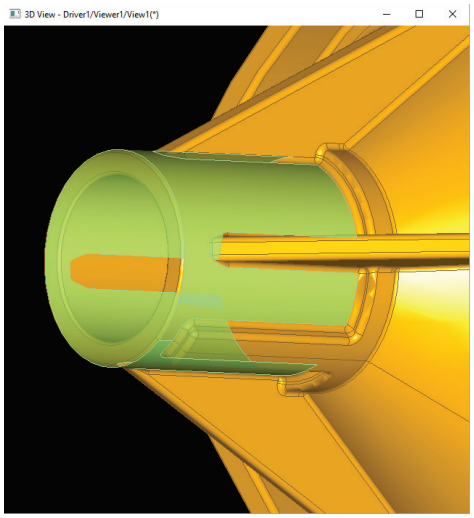
  

## Highlighting style

OCCT 7.1.0 improves look-and-feel of shaded dynamic highlighting of the object under the cursor in 3D viewer. Shaded highlighting now does not disable lighting.

Furthermore, highlighting API has been redesigned to be more customizable. Highlighting style is not just a color anymore - it is represented by new class `Graphic3d_HighlightStyle` and allows defining transparency and other properties.

`Graphic3d_HighlightStyle` is now a part of highlighting methods previously receiving color, including `AIS_InteractiveContext::HilightWithColor()`.

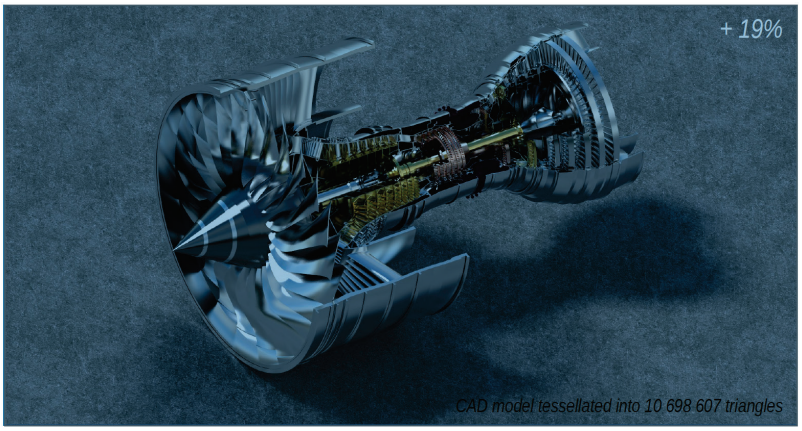



Shaded highlighting with transparency

## Ray-tracing renderer

Ray-tracing renderer has received several improvements.

Ray-Tracing now uses quad BVH (boundary volumes hierarchy), which improves the rendering performance and reducing GPU memory footprint.

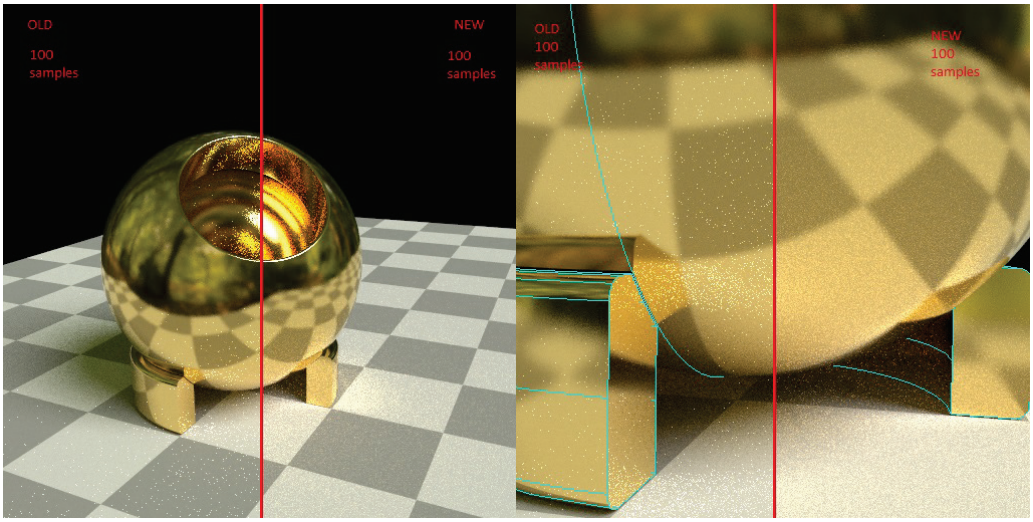



| BVH | QBVH |
|---|---|
| FPS (1280x720) | |
| 1.47 | 1.75 |
| Stack size | |
| 54 | 25 |
| Memory footprint | |
| 225.95 | 169.30 |

Physically-based material rendering within Path tracing engine has been adjusted for better alignment with reference data.

The rays generation for Path-tracing engine has been revised. New generator provides better quality (e.g. less noise) on complex scenes and requires less number of iterations.



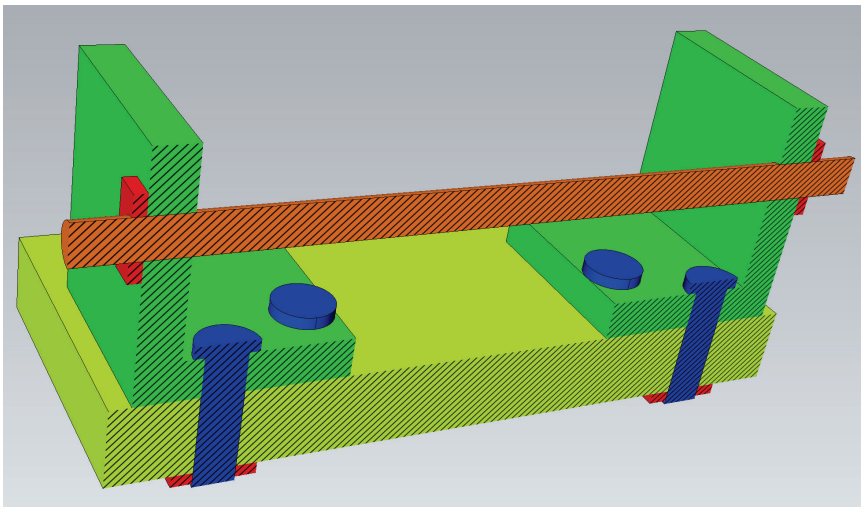

New adaptive screen space algorithm dramatically improves the efficiency of Path-Tracing algorithm by selectively optimizing the number of rays per screen region. This feature is currently available only on NVIDIA GPUs and managed by the flag `Graphic3d_RenderingParams::AdaptiveScreenSampling`.

## Clipping planes

Clipping planes functionality has been improved in this release.

Capping plane now can inherit material from the object, which is managed by new option `Graphic3d_ClipPlane::SetUseObjectMaterial()`.

Clipping and capping performance has been dramatically improved for scenarios when global plane is defined for entire view.



Global capping plane automatically inheriting material of sectioned assembly parts.

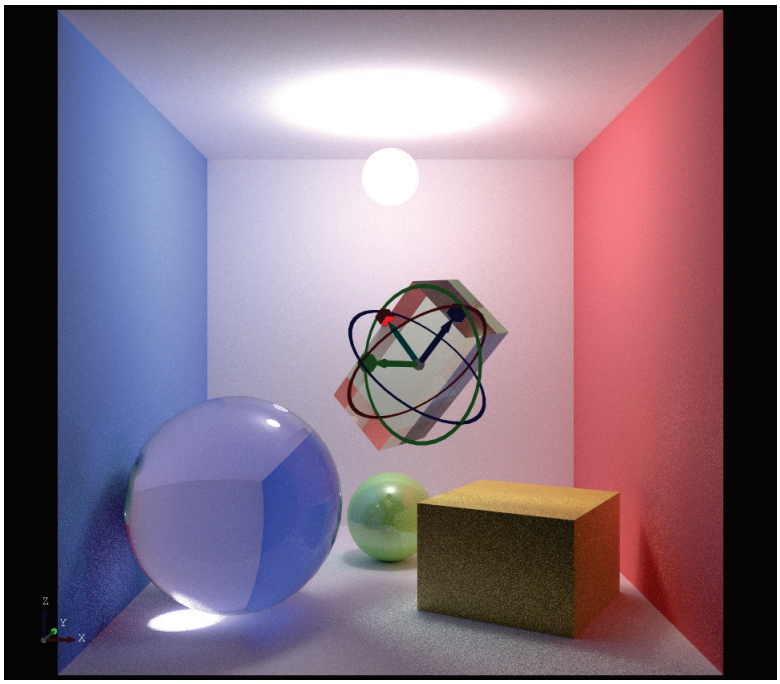
## Manipulator object

This release introduces new built-in interactive object `AIS_Manipulator` providing convenient interface for moving objects within 3D viewer. The class features three kinds of transformation modes for attached objects:

- translation through axis;
- scaling within axis;
- rotation around axis.

These modes can be enabled by the selection mode (from 1 to 3). There are three orthogonal transformation axes defined by position property of the manipulator. A particular transformation mode can be disabled for each of the axes or all of them. Furthermore each of the axes can be hidden or made visible.

The object can be activated using `vmanipulator` command within Draw Harness.



Manipulator object attached to the box

## Support of several attributes of one type at the same label

Several simple and most popular OCAF attributes: `TDataStd_AsciiString`, `TDataStd_Integer`, `TDataStd_Name` and `TDataStd_Real` have been extended.

Now several such attributes of one type may be placed at the same label using different user-defined GUIDs. For this, each attribute, which takes this custom GUID as an argument, now has new `Set` methods. For example:

```
const Standard_GUID MY_DENSITY ("5011c3e7-fa6f-4d27-be31-c83e70a4dc82");
…
TDataStd_Real::Set(aLabel, MY_DENSITY, 1.2);
```

This Integer attribute may be located at the same label with Integer with default GUID. This avoids creating a new sub-label to store an additional attribute of the same type.

The management of attributes on label remains the same in all other aspects.

The format version of OCAF persistent document has been incremented, thus files written by OCCT 7.1 will not be readable by previous versions of OCCT. Backward compatibility is preserved, i.e. previously saved documents are fully supported.

## *OCAF persistence without plugins*

The old dynamic loading mechanism of OCAF persistence based on usage of resource files and plugins is now deprecated. Instead, persistence drivers can be loaded directly in the application instance on C++ level. It is not necessary anymore to create dedicated class inheriting from `TDocStd_Application` and redefining its method `Formats()`. Class `TDocStd_Application` can be used in most cases.

Method `TDocStd_Application::DefineFormat()` allows defining format completely by a single call, including drivers to be used for persistence. Use it in your application to define supported formats, for instance:

```
myOcafApplication->DefineFormat ("NewDocumentFormat",
                                 "New format for OCAF documents", "ndf",
                                 new NewDocumentFormat_RetrievalDriver(),
                                 new NewDocumentFormat_StorageDriver());
```

Multiple formats can be defined, as before.

All OCAF driver packages provide static method `DefineFormat()` that defines standard OCAF persistence format supported by corresponding package. Use this method to enable support of corresponding format in your application, for instance:

```
BinOcaf::DefineFormat (myOcafApplication);
```

# Modifications

## Foundation Classes

| | |
|---|---|
| 24788 | *Summary:* Remove `Dico_Dictionary`.<br><br>Class `Dico_Dictionary` has been replaced by `NCollection_DataMap` and `NCollection_IndexedDataMap` classes. |
| 26380<br>27838 | *Summary:* Support `wchar_t*` input within `TCollection_AsciiString` and `TCollection_ExtendedString`.<br><br>The description of `TCollection_ExtendedString` and `TCollection_AsciiString` has been updated to reflect usage of these classes for Unicode strings. `TCollection_ExtendedString` now defines constructor taking `wchar_t*` (all platforms) and method `::ToWideString()` returning `wchar_t*` (Windows only). `TCollection_AsciiString` now defines constructor taking `wchar_t*`.<br><br>New methods have been added in classes `TCollection_ExtendedString` and `TCollection_AsciiString`:<br>▪ `StartsWith()` determines whether the beginning of this string instance matches the specified string.<br>▪ `EndsWith()` determines whether the end of this string instance matches the specified string.<br><br>Other modifications:<br>▪ `TCollection_ExtendedString` internals has been updated to eliminate duplicated code for converting between UTF-16 and UTF-8.<br>▪ Global methods `OSD_OpenStream()` and `OSD_OpenFileBuf()` have been replaced by C++ template to eliminate copy-paste for different STL collections.<br>▪ `OSD_SharedLibrary` now uses wide-char system API call `LoadLibraryExW()` on Windows for consistency.<br>▪ New macro `Standard_UNUSED` has been added for marking possibly unused functions and variables, to prevent redundant compiler warnings (GCC and Clang) when template specialization is used. |
| 27432 | *Summary:* Null Handle access when creating `AIS_Trihedron`.<br><br>Debug-only `RaiseIf` has been replaced by a simple `if` in `Units_Sentence.cxx` to correctly handle an exceptional situation in Release mode. |
| 27522 | *Summary:* `FSD_BinaryFile` - incorrect size check in a stream.<br><br>`FSD_BinaryFile::ReadExtendedString()` now compares read bytes instead of symbols. |
| 27563 | *Summary:* `opencascade::handle` - make `operator*()` consistent with `operator->()`.<br><br>`Const operator*()` and method `get()` now return non-const pointer for consistency with `operator->()`. |
| 27602 | *Summary:* `gp_TrsfForm.hxx` - wrong comment for enumeration.<br><br>Comments have been corrected in `gp_Trsf.hxx`. |

| | |
|---|---|
| 27675 | *Summary:* Handle Unicode path to `CSF_UnitsLexicon` and `CSF_UnitsDefinition` on Windows.<br><br>Unicode-aware functions `OSD_OpenStream` and `OSD_FileStatCTime` are used in `Units` package for fetching file timestamp. |
| 27706 | *Summary:* `OSD_Path::DownTrek()` crashes on empty argument.<br><br>Check on empty argument has been added in `OSD_Path::DownTrek` to avoid crash. |
| 27772<br>27676 | *Summary:* Define `Standard_Boolean` using C++ type `bool` instead of `unsigned int`.<br><br>The following type definitions in OCCT have been modified to use C++11 types:<br>▪ `Standard_Boolean` is now `bool` (previously `unsigned int`).<br>▪ `Standard_ExtCharacter` is now `char16_t` (previously `short`).<br>▪ `Standard_ExtString;` is now `const char16_t` (previously `const short`).<br>▪ `Standard_Utf16Char` is now `char16_t` (previously `uint16_t` for compatibility with old compilers).<br>▪ `Standard_Utf32Char` is now `char32_t` (previously `uint32_t` for compatibility with old compilers).<br>For most applications this change should be transparent on the level of source code. Binary compatibility is not maintained, as bool has different size in comparison with unsigned int.<br><br>The code has been updated to remove no-op casts and implicit casts to `Standard_Boolean` as well as improper use of `Standard_Boolean` instead of `Standard_Integer`:<br>▪ Bit flags are defined as private enumerations in classes `Bnd_Box` and `Bnd_Box2d`.<br>▪ Bit flags are defined as enumerations in classes `HLRAlgo_BiPoint`, `HLRAlgo_EdgesBlock`, `HLRBRep_EdgeData` and `HLRBRep_FaceData`.<br>▪ Bit flags are defined as Boolean fields in classes `HLRAlgo_EdgeStatus`, `HLRBRep_BiPnt2D` and `HLRBRep_BiPoint`.<br>▪ Bit flags are defined as `Standard_Integer` in `HLRAlgo_PolyData`.<br>▪ Boolean flag is now defined as `Standard_Boolean` in `OSD_DirectoryIterator` and `OSD_FileIterator`.<br>▪ `ShapeAnalysis_Surface::SurfaceNewton()` now returns `Standard_Integer` (values 0, 1 or 3).<br>▪ `ChFi2d_FilletAlgo` now uses `TColStd_SequenceOfBoolean` instead of `TColStd_SequenceOfInteger` for storing Boolean flags.<br><br>Other modifications:<br>▪ Method `IFSelect_Dispatch::PacketsCount()` has been dropped from the interface.<br>▪ `ShapeFix_Solid::Status()` has been fixed to decode requested status instead of returning integer value.<br>▪ `TopOpeBRepBuild_Builder1` now defines map storing `Standard_Boolean` values instead of `Standard_Integer`.<br>▪ Persistence for `Standard_Boolean` type has been corrected to keep backward compatibility: `BinMDataStd`, `BinTools`, `FSD_BinaryFile`.<br>▪ Broken Draw Harness commands `vdisplaymode` and `verasemode` have been removed.<br>▪ Method `BRepMesh_FastDiscretFace::initDataStructure()` provides a workaround for old GCC limitations<br>▪ Ambiguity is avoided in method `BRepMesh_IncrementalMesh::clear()`. |

| | |
|---|---|
| 27794 | *Summary:* A danger in the `NCollection_Vector` initializing constructor.<br><br>Constructor of `NCollection_Vector` is now declared as explicit. |
| 27810 | *Summary:* Compilation error on `NCollection_UBTree::Kill()`.<br><br>The type of passed parameter `(Handle(NCollection_BaseAllocator)& theAlloc` is correctly indicated as `const` to avoid compilation error. |
| 27844 | *Summary:* Remove method `OSD_File::Print()`.<br><br>Redundant method `OSD_File::Print()` has been removed. |
| 27849 | *Summary:* `ResourceManager` path computations fail for the folders containing dots.<br><br>`Resource_Manager` has been corrected to properly handle paths with dots inside.<br><br>Extraction of extension from path in DOS mode when working on Linux has been corrected in `OSD_Path`. |
| 27854 | *Summary:* `OSD_Parallel::NbLogicalProcessors()` should handle Android specifically.<br><br>The method reading number of available CPUs uses sysfs on Android to return the number of physically available cores (instead of the number of currently active cores as earlier). |
| 27901 | *Summary:* `Resource_Manager` creates a directory with insufficient permissions.<br><br>`Resource_Manager` now sets RWXD permissions for the created directories to allow their removal. Method `OSD_Directory::Build()` will create intermediate directories if they do not exist. |
| 27915<br>27941 | *Summary:* Remove the class `NCollection_QuickSort`.<br><br>Class `NCollection_QuickSort` has been removed. It is recommended to use sorting algorithms provided by STL instead. |
| 28040 | *Summary:* `Message_Printer` - handle `Standard_CString` messages as UTF-8.<br><br>Strings passed to printer in the form of `Standard_CString` are now assumed to be in UTF-8 encoding. |
| 28103 | *Summary:* Extend `NCollection_Map` to check whether two given maps are intersected.<br><br>New method `NCollection_Map::HasIntersection` checks whether two given maps contain at least one same item.<br>The corresponding command `HasIntersection` has been added in Draw. |

## Application Framework

| | |
|---|---|
| 23639<br>25812 | *Summary:* Replace dynamic loading mechanism of OCAF persistence with dynamic-link one.<br><br>The old dynamic loading mechanism of OCAF persistence based on usage of resource files has been replaced by ordinary dynamic link one. See the details in New Features section. |
| 25534<br>28039 | *Summary:* `TObj_Application` Unicode path issue.<br><br>Methods `TObj_Model::SaveAs` and `TObj_Model::Load` now receive `TCollection_ExtendedString` filename arguments instead of `char*`. |
| 26832 | *Summary:* `TFunction_Iterator` will not work if `ExecutionStatus` is `Standard_False`.<br><br>The method `TFunction_Iterator::More()` has been corrected to enter loop properly. |
| 27192 | *Summary:* Improvement of storage of OCAF document in XML file format.<br><br>Writing of data of some OCAF attributes in XML file format has been improved:<br>▪ `TDataStd_ExtStringArray` uses separators to distinguish strings written in line (instead of writing one line per string).<br>▪ A default GUID obtained by the method `GetDefaultTreeID()` is now skipped in the written XML file. |
| 27433 | *Summary:* Opening/Saving OCAF Documents With Own Document Format Does Not Work.<br><br>The method `Resource_Manager::Load()` is corrected. An empty file name is no more appended to the directory name, since this is not necessary and causes `Standard_OutOfRange` failure in `TCollection_AsciiString`. Only non-empty file names are appended. |
| 27454 | *Summary:* Application hangs while opening a non-OCAF XML file.<br><br>Protection against accessing null pointer has been added in `PCDM_ReadWriter::FileFormat`.<br><br>`LDOM_XmlReader::ReadRecord` now takes into account that the character '>' may have no special meaning (e.g., in a text), thus we must read the data behind this character to complete the current tag reading. |
| 27604 | *Summary:* Memory is not released after closing XBF file.<br><br>Virtual method `BinLDrivers_DocumentRetrievalDriver::Clear` now allows clearing the accumulated cache data when it is not needed anymore, e.g. `BinMNaming_NamedShapeDriver`, a set of stored shapes. |
| 27619 | *Summary:* Improvement of data manipulation by `TPrsStd_AISPresentation` attribute.<br><br>The method `TPrsStd_AISPresentation::getData()` now calls `FindAttribute()`; `myData` field has been removed. |

| | |
|---|---|
| 27726 | *Summary:* List of formats supported by `TDocStd_Application`.<br><br>New methods `TDocStd_Application::ReadingFormats` and `TDocStd_Application::WritingFormats` return a list of supported formats for retrieving and storing documents. |
| 27821<br>27835<br>27991 | *Summary:* `BinXCAF` - handle correctly faces with `NULL` surface within `BinTools_ShapeSet`<br><br>`BinTools_ShapeSet::AddGeometry()` now writes `NULL` surface with 0 index. `WithTriangles` option is ignored for `TopoDS_Face` with `NULL` surface so that triangulation-only faces are not lost with default options.<br><br>New Draw Harness command `StoreTriangulation` defines `WithTriangulation` flag for `BinXCAF/BinOcaf` storage drivers.<br><br>`BRepGProps` now ignores faces without geometric surface to avoid access violation.<br>`BRepExtrema_DistShapeShape::DistanceMapMap()` now skips comparison between void bounding boxes.<br>`BRepBndLib::Add()` now ignores `useTriangulation` flag for faces without geometric surfaces, and uses triangulation if any for updating of the box. |
| 27932<br>28134 | *Summary:* Improvement of standard attributes usability<br><br>OCAF attributes TDataStd_AsciiString, TDataStd_Integer, TDataStd_Name and TDataStd_Real have been extended with the possibility to use custom GUID.<br><br>See the details in New Features section. |
| 28034 | *Summary:* Stack overflow in LDOM destructor.<br><br>Recursion is replaced by iteration in destructor of the class `LDOM_MemManager::MemBlock` to avoid stack overflow. |

## Modeling Data

| | |
|---|---|
| 25649<br>27108 | *Summary:* `GCPnt_TangentialDeflection` does not respect linear deflection.<br><br>`GCPnt_TangentialDeflection` algorithm prevents violation of angular and curvature deflection condition for smooth curve intervals.<br><br>The algorithm for calculation of maximal deflection has been modified in Draw commands `crvtpoints` and `crvpoints` |
| 27264 | *Summary:* Weird difference between two `BRepTools::Write()` overloads.<br><br>DRAW command `restore` now assumes that the file contains shape in BRep format by default, and thus can read files without `DBRep_DrawableShape` header in the first line. |
| 27491 | *Summary:* Document thread-safety behavior of `GeomAdaptor_Curve.`<br><br>It is explicitly stated in the comments of `Adaptor` classes that they are not thread safe by design. |
| 27703 | *Summary:* `BinTools_ShapeSet` stream read bug.<br><br>Reading and writing out of array bounds is now avoided in method `BinTools_ShapeSet::Read`. |
| 27863 | *Summary:* `Geom_BSplineSurface::SetVKnot(const Standard_Integer, const Standard_Real)` has no effect.<br><br>The function `Geom_BSplineSurface::SetVKnot` now has the same behavior as `SetUKnot`. |

## Modeling Algorithms

| | |
|---|---|
| 23178<br>26894<br>27856<br>28009 | *Summary:* Intersection of cylinders fails to produce results.<br><br>The following modifications have been introduced to improve processing of cylinders intersection:<br><br>▪ The interface of method `IntPatch_ImpImpIntersection::Perform()` has been changed.<br>▪ New method `IntPatch_ImpImpIntersection::GetStatus()` provides status information and makes intersection algorithm more informative and flexible for using.<br>▪ New class `Bnd_Range` describes a range in 1D space restricted by two real values. A range can be void indicating there is no point included in the range.<br>▪ `AddBoundaryPoint` function obtains intersection points in both boundaries (`VFirst` and `VLast` of every surface).<br>▪ The increase of B-spline degree value resulting from `Geom2dConvert::ConcatG1()` has been limited by `Geom2d_BSplineCurve::MaxDegree()` value (max degree = 25).<br>▪ The algorithm of B-spline closure definition has been changed in methods `Geom2dConvert::C0BSplineToC1BSplineCurve()` and `Geom2dConvert::C0BSplineToArrayOfC1BSplineCurve()`.<br>▪ The method `IntPatch_Intersection::GeomGeomPerfomTrimSurf()` has been removed due to unification of trimmed and not-trimmed cylinders processing. |
| 24056 | *Summary:* Error LNK2019: unresolved external symbol.<br><br>Unused private fields have been removed from some classes in `TopOpeBRepDS` |
| 25214 | *Summary:* HLR fails to project edges<br><br>The classes `HLRBRep_Data` and `HLRBRep_InternalAlgo` now avoid getting non-existing array items. Reference to array items is used instead of pointers where possible. |
| 25957 | *Summary:* `nurbsconvert` modifies original shape.<br><br>`BRepTools_NurbsConvertModification` algorithm has been improved to ensure that the subshapes from the original shape are not changed by the operation.<br><br>See the details in New Features section. |
| 26270 | *Summary:* `GeomFill_NSections` constructor crash.<br><br>The class `GeomFill_NSections` has been modified to avoid undefined associated parameters causing exception. |

| | |
|---|---|
| 26329<br>27713<br>27842<br>27891<br>27892<br>27929 | *Summary:* Restore floating point signals handling in DRAW.<br><br>New DRAW command `dsetsignal` resets `OSD` signal handler with either armed or disabled FPE handler, according to an option. If called without arguments, this command sets FPE handler only if environment variable `OSD_FPE` is defined (with value different from 0).<br>On start, DRAW calls `dsetsignal` to set FPE signal if `CSF_FPE` is defined.<br><br>The following changes have been made to fix floating point exceptions arising after enabling signals:<br>▪ Global functions `Sinh()` and `Cosh()` defined in `Standard_Real.hxx` raise `Standard_NumericError` exception if the argument is too big (greater than 710.47586), instead of relying on system treatment of floating point overflow. These functions are used instead of sinh and cosh in `ElCLib.cxx`.<br>▪ Maximal value of parameter on hyperbola is restricted by 23 (corresponding to ~1e10 in 3d) in order to avoid FP overflow in `Extrema_GenExtCS.cxx` and `ShapeFix_EdgeProjAux.cxx`.<br>▪ Interface of the root curve adaptor class `Adaptor3d_Curve` has been updated with new virtual methods `BasisCurve` and `OffsetValue`. They complement the adaptor for the case of offset curves. These methods are used in Extrema_GenExtCS.cxx to restrict domain search in the case of offset of hyperbola, in order to get rid of floating point overflow. All classes inheriting `Adaptor3d_Curve` have been changed to implement new virtual methods.<br>▪ Protection against division by zero has been implemented in `ApproxInt_KnotTools.cxx`, `BRepClass3d_SClassifier.cxx`, `BRepGProp_Face.cxx`, `BRepMesh_FastDiscretFace.cxx`, `Geom2dGcc_Circ2d2TanOnIter.cxx`, `Geom2dInt_Geom2dCurveTool.cxx` and `IntPolyh_MaillageAffinage.cxx`.<br>▪ Protection against calling math functions of infinite arguments has been added in `BRepCheck_Edge.cxx`, `BRepLib.cxx`, `CSLib_NormalPolyDef.cxx`, `Extrema_FuncExtPC.gxx`, `Extrema_GextPC.gxx`, `Extrema_GlocateExtPC.gxx`, `Intf_InterferencePolygonPolyhedron.gxx`, `ShapeAnalysis_Surface.cxx`, `ShapeAnalysis_TransferParametersProj.cxx`, `ShapeAnalysis_Wire.cxx` and `math_FunctionSetRoot.cxx`.<br>▪ Proper initialization of local variables is implemented in `BOPAlgo_PaveFiller_6.cxx` and `XSDRAWSTLVRML.cxx`.<br>▪ Inconsistent usage of `Standard_Boolean*` to access integer data in HLR has been corrected<br>▪ The field `Geom_OffsetSurface::myOscSurf` is properly initialized if the offset of a trimmed BSpline surface is created.<br>▪ The osculating surface is not used in method `GeomEvaluator_OffsetSurface::CalculateD1` if the normal to basis (for offset) surface is well defined.<br>▪ Interfaces of methods `DistanceMinimizeByGradient` and `DistanceMinimizeByExtrema` from class `IntWalk_PWalking` have been improved. |
| 26738 | *Summary:* Improve safety of processing arguments in Boolean operations when running with fuzzy option<br><br>The tolerance of input shapes is not increased when fuzzy option is in force. Instead the tolerances of sub-shapes are increased by fuzzy value everywhere where it is needed by the intersection algorithms.<br><br>See the details in New Features section. |

| | |
|---|---|
| 26747 | *Summary:* Some constructors of `gp_Parab2d` class contain redundant parameters.<br><br>Unused constructors have been removed from class `gp_Parab2d`. The method `gp_Parab2d::Directrix()` now avoids reversing the directrix. |
| 26917<br>27139<br>28046<br>28050 | *Summary:* 3D Offset algorithm produces incorrect result.<br><br>3D Offset algorithm supports new configurations of the argument shape of the Offset operation for mode `Complete` and `Joint type` intersection of the shapes.<br><br>See the details in New Features section. |
| 26938 | *Summary:* Boolean operations fail between two ellipsoids.<br><br>New function `HandleSingleSingularPoint` from class `IntWalk_PWalking` provides processing for a single singular point. |
| 26972<br>27177 | *Summary:* `BRepClass3d_SolidClassifier` doesn't take into account vertex/edge/face tolerances.<br><br>3D point-solid classifier now takes into account vertex/edge tolerances. If the given point lies inside the tolerance area of a vertex or edge of the solid it is classified as `TopAbs_ON`.<br><br>`IntCurvesFace_Intersector::Perform` can use an optional null-tolerance to classify a 2d-point relatively to the given face. `UBTreeFiller` is used to speedup intersection process between edges/vertices and the point.<br><br>Bounding box tree functionality is now provided by a separate class `BRepClass3d_BndBoxTree`. |
| 27015 | *Summary:* Sewing returns invalid shape if some faces are nearly plane cones.<br><br>The interpolation state check in class `Approx_SameParameter` marks as valid only ordered sequences of poles. |
| 27033 | *Summary:* Ensure stable result of area calculation.<br><br>The method `BRepGProp_Face::LKnots` has been corrected to properly initialize the output data in case of circle. |
| 27082<br>27309 | *Summary:* `UnifySameDomain` must add internal edges where appropriate to resolve self-intersections.<br><br>The new option `AllowInternalEdges` in class `ShapeUpgrade_UnifySameDomain` defines how to treat two faces that can be merged together but are connected with another face via the common edge. When this option is `true`, the faces are merged and multi-connected edges are added to the merged face as `INTERNAL` edges. |
| 27119 | *Summary:* `Unifysamedomain` regression issue in OCCT 7.<br><br>`UnifySameDomain` algorithm now can process and unify compounds (in addition to faces from shells). |

| | |
|---|---|
| 27151 | *Summary:* Exception is raised during performing command `splitshape` in the Test Harness.<br><br>Protection against overlapping edges has been added in various classes from `LocOpe` package. |
| 27184 | *Summary:* `BRepExtrema_DistShapeShape` returns wrong result.<br><br>The local optimization default algorithm of class `Extrema_FuncDistSS` is now distance-based. |
| 27194 | *Summary:* Possible division by zero in `IntPatch_WlineTool`.<br><br>Correct handling for division by zero has been added in `IntPatch_WlineTool` class. This prevents exception when FPE is enabled. |
| 27222 | *Summary:* Empty result of making volume operation.<br><br>`BOPAlgo_MakerVolume` takes into account the possibility of finding Same Domain faces while collecting faces for building solids.<br><br>`BOPDS_DS::CheckCoincidence` now adds `Precision::Confusion()` to the intersection tolerance. |
| 27234 | *Summary:* Code duplication: `Convert_CompBezierCurvesToBSplineCurve*` in `ShapeConstruct`.<br><br>▪ Some classes from `ShapeConstruct` package have been removed as duplicates of the ones from `Convert` package.<br>▪ `Convert` classes have been protected against joining segments when the degree is 1.<br>▪ Optimization previously made in `Convert_CompBezierCurvesToBSplineCurve` class has been applied to 2d equivalent. |
| 27269 | *Summary:* Intersection algorithm produces null-length curve.<br><br>New function `isDegenerated` from `GeomInt_IntSS` class checks if the restriction line corresponds to a degenerated edge. In this case a null-curve is returned instead of null-length curve. |
| 27271 | *Summary:* `Unifysamedomain` invalid result.<br><br>The algorithm `ShapeUpgrade_UnifySameDomain` has been modified to avoid merging faces along periodic direction. |
| 27280 | *Summary:* HLR algorithms taking seam edges into account.<br><br>`HLRAppli_ReflectLines` algorithm now filters out seam edges in the results.<br>New Draw command `hlrin3d` has been added. |
| 27299 | *Summary:* Incorrect result of the normal projection algorithm.<br><br>Functions `GetContinuity()` and `SetContinuity()` that get and set the continuity of local border splits have been added in algorithm `math_GlobOptMin` to handle the case of a seam edge passing by the initial curve. |

| | |
|---|---|
| 27300 | *Summary:* Boolean operation produces invalid shape in terms of `bopargcheck` command.<br><br>It is now checked if the value found by `math_PSO` algorithm can be precised by `math_NewtonMinimum` algorithm. If not, `math_PSO` algorithm is called again with different parameters.<br><br>New method `math_NewtonMinimum::GetStatus()` returns the computation `Status`. |
| 27302 | *Summary:* Invalid curves number in intersection result.<br><br>The method `ElCLib::InPeriod()` has become faster and more reliable in frame of processing `FLT_OVERFLOW` and `DIVISION_BY_ZERO` cases. |
| 27310 | *Summary:* Huge tolerance obtained in the result of intersection of two cylindrical faces.<br><br>New function `BoundariesComputing` from `IntPatch_ImpImpIntersection` class computes true domain of future intersection curve, which avoids excess iterations to determine intersection. The use of `CylCylMonotonicity()` function has been improved. |
| 27315 | *Summary:* UnifySameDomain leaves unmerged edges.<br><br>The internal function `MergeEdges` from `ShapeUpgrade_UnifySameDomain` class has been adapted to the case when it takes on input a set of edges constituting several connected chains. |
| 27322<br>28054 | *Summary:* `geom/revolution_00/A1`: Incorrect pcurve creation.<br><br>The algorithm calculating projection line on cone has been corrected in method `ProjLib_Cone::Project`. The special case when the starting point of the line coincides with the cone apex is fixed by shifting this point along the source line.<br><br>The method `GeomInt_IntSS::BuildPCurves()` now adjusts first or last knots of a 2D Curve.<br><br>Projection of line on cone has been improved in method `ProjLib_ComputeApprox::Function_SetUVBounds()`. |
| 27340<br>27341<br>27557 | *Summary:* Incorrect exact HLR results.<br><br>The following improvements have been introduced in HLR algorithm:<br><br>▪ Excess interferences in case of simple hiding face have been removed<br>▪ Processing of boundary edges coinciding with outlines has been corrected<br>▪ The number of samples in `PolyPoly` intersection algorithm tuned in each instance of generic class.<br>▪ Default number of the samples in `Geom2dAdaptor` corresponds to the `Adaptor2d_Curve2d` class.<br>▪ The minimal number of B-Spline points is changed.<br>▪ The minimal number of supported samples has been added in intersection algorithms. |

| | |
|---|---|
| 27352 | *Summary:* Optimal axis-aligned bounding box for a shape.<br><br>New method `BRepBndLib::AddOptimal()` builds a precise bounding box, which differs from the exact geometry boundaries of the shape only by the tolerances of shape entities. This algorithm is the same as for method `AddClose()`, but uses more precise methods for building boxes of geometric objects. |
| 27357 | *Summary:* `Geom2dGcc_Circ2d2TanOn`: check status of sub-algorithms to avoid exceptions.<br><br>Additional checks for underlying algorithms have been added in some methods of class `Geom2dGcc_Circ2d2TanOn`. |
| 27368<br>27634 | *Summary:* Finding objects in vicinity of a ray.<br><br>Radix sort functionality from `BVH_LinearBuilder` has been generalized and implemented as separate classes in `BVH` package. The basic API of sorting class is given in `BVH_Sorter` class, while `BVH_QuickSorter` and `BVH_RadixSorter` provide quick sorting and radix sorting algorithms. |
| 27371 | *Summary:* `BRepExtrema` works much slower.<br><br>The computation of Lipschitz constant has been improved in `Extrema_GenExtCC` class. The class `math_GlobOptMin` has been refactored. |
| 27386 | *Summary:* `BRepOffsetAPI_MakePipeShell` does not provide history of generations<br><br>The method `BRepFill_PipeShell::BuildHistory` has been redesigned: now it builds generated shapes for sub-edges and sub-vertices of sections. |
| 27391 | *Summary:* `BRepLib::EnsureNormalConsistency()` raises exception in case of asynchronous `PolygonOnTriangulation` problem.<br><br>Additional check have been added to `BRepLib::EnsureNormalConsistency()` function: any edge with two adjacent faces will be skipped (while updating the normals) if the number of discretization points (nodes) is different for each face on this edge. |
| 27428<br>27441<br>27896<br>28076 | *Summary:* The method `IntTools_Context::IsVertexOnLine` incorrectly computes parameter of the point on the curve.<br><br>The method `IntTools_Context::IsVertexOnLine`, which computes the point parameter on curve, now chooses the closest bound to the point.<br><br>In `IntPatch_PrmPrmIntersection` algorithm, `Purger` is disabled if some points have been added in the Walking line. |
| 27431 | *Summary:* Huge tolerance obtained during intersection of cylinder and sphere.<br><br>The processing of cases when the intersection line goes through the apexes of a sphere has been improved. |
| 27448<br>27519 | *Summary:* `BOPTools_AlgoTools::IsMicroEdge` does not correspond to shape validity criteria.<br><br>The treatment of shape validity criteria has been improved. |

| | |
|---|---|
| 27466 | *Summary:* The algorithm `Extrema_GenLocateExtPS` gives incorrect result.<br><br>New class `Extrema_FuncPSDist` implements Euclidean distance criteria to search for local point / surface extrema. |
| 27467 | *Summary:* Class `Extrema_ExtCC2d` does not find the extremum between two intersecting lines.<br><br>Line / line analytic treatment has been added in function `Extrema_ExtElC2d`. |
| 27468 | *Summary:* Incorrect processing of some cases by HLR algorithm.<br><br>HLR algorithm has been improved by correcting usage of tolerances in `HLRBRep_Intersector::Perform`, 2d curve sampling in `HLRBRep_CurveTool::NbSamples`, etc. |
| 27475 | *Summary:* Incomplete direction for Powell method in `math_GlobOptMin`.<br><br>Directions in method `math_GlobOptMin::computeLocalExtremum` now form a valid orthogonal set. |
| 27481<br>27537 | *Summary:* `GCPnts_TangentialDeflection` produces incorrect number of sample points for circular edge.<br><br>The number of sample points for circular edge is rounded up in method `GCPnts_TangentialDeflection::PerformCircular` to satisfy `curvatureDeflection` more precisely. |
| 27493<br>27565 | *Summary:* `Extrema_ExtCC` does not set flag `IsParallel` equal to true for the overlapped curves.<br><br>`Extrema_GenExtCC::Perform()` algorithm has been improved to check for infinite solutions starting with two (instead of 100). |
| 27520 | *Summary:* Operation `splitshape` in the Test Harness gives invalid result on the attached case.<br><br>The new method `BRepFeat_SplitShape::Add` adds a sequence of splitting edges or wires for the whole initial shape without specification of the edge->face, edge->edge mapping. |
| 27521 | *Summary:* `Standard_ConstructionError` when trying to use `unifySameDom` into a shape.<br><br>The function `MergeEdges` from `ShapeUpgrade_UnifySameDomain` class has been corrected to take into account the orientation of the next edge added to the chain. |
| 27540 | *Summary:* Run-to-run differences in the 3D Offset algorithm.<br><br>The offset vertices are now calculated in method `BRepOffset_Inter2d::FuseVertices()` by superposition of intersection vertices between pairs of edges. To properly calculate the superposition of vertices they are sorted by method `BOPTools_AlgoTools::MakeVertex()`. |

| | |
|---|---|
| 27544 | *Summary:* Compiling OCCT with gcc version 4.8.2 gives error “array subscript is above array bounds”.<br><br>Possible out of borders problem has been fixed in method `IntPatch_InterferencePolyhedron ::TangentZoneValue`. |
| 27552 | *Summary:* Wire creation fails depending on the order of edges.<br><br>Method `BRepBuilderAPI_MakeWire::Add()` now takes into account the order of edges and the geometric proximity of free vertices to the existing wire and the input edges. Coincident free vertices are fused into one. The original wire remains untouched topologically (yet the tolerances and points can be modified). |
| 27568 | *Summary:* Exception is thrown when a degenerated edge is added to a wire.<br><br>Methods from `BRep_Tool` package have been corrected to ensure that output arguments are always initialized. |
| 27569 | *Summary:* Projecting a curve hangs.<br><br>Parameterization speed is taken into account in `ShapeConstruct_ProjectCurveOnSurface` class to avoid `ProjLib` projector usage in case of bad input data. |
| 27581 | *Summary:* Logical error in `Bnd_Box(2d)::SquareExtent()` method.<br><br>Local variables `dx` and `dy` in method `Bnd_Box(2d)::SquareExtent()` take into account the field “Gap” twice. |
| 27664 | *Summary:* Incomplete intersection curve from the attached shapes.<br><br>Computation of offset values in method `IntPatch_RstInt::PutVertexOnLine` has been corrected to provide correspondence and adjustment to periods between Domain of WLine and surface domain. |
| 27677 | *Summary:* Incorrect CUT of a solid by semi-infinite solid.<br><br>`ProjLib_Cylinder` now returns isDone=false if the projected line is not parallel to cylinder axis or if the plane of the projected circle is not orthogonal to cylinder axis |
| 27679 | *Summary:* Wrong offset: overlapping edges.<br><br>The processing of closed bisectors has been corrected in method `BRepFill_OffsetWire::UpdateDetromp()`. |
| 27704 | *Summary:* Numeric inaccuracy due to huge extension of the offset faces.<br><br>The maximum extension value of faces in offset operation `BRepOffset_Tool::EnLargeFace` has been decreased to 1.e+7 to improve calculation precision. |

| | |
|---|---|
| 27719<br>27720<br>28053<br>28081 | *Summary:* HLRBRep_Algo incorrect output.<br><br>Intersection of curves and surfaces in HLRBRep uses parametric limits of face instead of parametric limits of surface. Method Contap_Contour::ComputeCloseLine is used in all cases when it is necessary to build an outline.<br><br>The algorithm building an outline in 2d parametric space of the surface now can better process complex surfaces with breaks and holes. |
| 27761 | *Summary:* Infinite loops intersection.<br><br>Incorrect adjustment has been removed from the static method IntTools_FaceFace::CorrectSurfaceBoundaries() used for correction of face boundaries before their intersection. |
| 27762 | *Summary:* Incorrect result of General Fuse operation.<br><br>The algorithm finding extrema solutions of a point and a torus in Extrema_ExtPElS has been corrected for the case of the major torus radius equal to zero. |
| 27766 | *Summary:* Incorrect section curves between attached cylinders.<br><br>The maximum 2d-tolerance has been limited to provide precise computation in method IntPatch_ImpImpIntersection::Perform(). |
| 27769 | *Summary:* BRepOffsetAPI_MakePipeShell produces a face based on degenerated toroidal surface<br><br>Method BuildKPart from GeomFill_Sweep class creates a sphere instead of torus if the major radius is less than tolerance. |
| 27773 | *Summary:* Empty result of section operation between line and offset of a circle.<br><br>The Resolution for Offset Curve based on an elementary curve (Line, Circle or Ellipse) is computed in class IntTools_EdgeEdge using this elementary curve. |
| 27774 | *Summary:* Constructor GeomPlate_BuildAveragePlane crashes if two input normals are parallel to each other.<br><br>GeomPlate_BuildAveragePlane algorithm now checks if the input normals are valid. |
| 27775 | *Summary:* Different behavior of GeomFill_BSplineCurves algorithm in DEBUG and RELEASE mode.<br><br>GeomFill_BSplineCurves::Init throws an exception in case of incorrect input data. |
| 27780 | *Summary:* Face-face intersection produces 2D curve that has reversed derivative at its end.<br><br>The API of method IntPatch_WLineTool::ComputePurgedWLine has been changed to insert a new Boolean parameter RestrictLine. If this parameter is false, the removal of outside points is skipped, and the result line is not distorted. This flag is determined inside IntTools_FaceFace to tell if it is necessary to limit the intersection line by surface domain. |

| | |
|---|---|
| 27804 | *Summary:* Two BReps cause intersections to loop for too long/infinitely.<br><br>In `ProjLib_ComputeApprox` algorithm, correct parametric tolerance is computed from the input 3D tolerance using surface resolution, in order to pass it to low-level 2D algorithm `Approx_FitAndDivide2d` (instantiation of the generic class `Approx_ComputeCLine`). Earlier 3D tolerance was used as parametric tolerance directly, which was a problem for surfaces with too small radius of curvature.<br><br>Redundant creation of type `AppParCurves_MultiCurve` object at iteration of method `Approx_ComputeCLine::Compute` has been eliminated.<br><br>The post treatment of Edge/Edge intersections has been improved in the following way:<br><br>▪ The procedure of sharing Edge/Edge intersection vertices has become consistent with intersection algorithm by enlarging the bounding box of each vertex by half of `Precision::Confusion()`;<br>▪ The vertex tolerance computation algorithm has been changed for Line/Circle cases to cover the tangent zone between Edges. |
| 27822 | *Summary:* Exception access violation is raised in `BRepOffsetAPI_MakePipeShell` during build.<br><br>New public method `IsDone()` is added to abstract class `BRepFill_SectionLaw`. |
| 27830 | *Summary:* Infinite HLR looping.<br><br>New function `IsBadFace` from `HLRBRep_Data` class protects HLR algorithm against incorrect input faces (e.g. with U bounds exceeding the period in thousands times). In addition, the method `GeomInt::AdjustPeriodic` is now used to fit the intersection point in the period for periodical faces. |
| 27862 | *Summary:* Exception in `BRepOffsetAPI_MakePipeShell`.<br><br>`BRepFill_Sweep` algorithm checks now if the shape is not null before querying its `ShapeType`. |
| 27870 | *Summary:* Refactoring of HLR algorithms<br><br>The HLR algorithms from `TKHLR` toolkit have been refactored to use typed data structures and access them by names instead of using low-level types `Standard_Address` and macros involving low-level casts to access particular data elements. |
| 27873 | *Summary:* Exception is raised in `BRepFill_Filling::FindExtremitiesOfHoles()`.<br><br>Checks and warnings about incorrect input objects have been added in class `GeomPlate_BuildPlateSurface`. |
| 27875 | *Summary:* `GeomFill_NSections` constructor crash on sequence of curve containing only one curve.<br><br>A warning that `GeomFill_NSections` algorithm cannot create surface from a sequence with only one curve has been added. |

| | |
|---|---|
| 27888 | *Summary:* Fuse of valid untouched solids leads to result with faults.<br><br>Classification of a point relatively to a solid that led to faulty interferences between vertices/edges of solids has been corrected in function `ClassifyUVPoint` from class `BRepClass3d_SolidExplorer` by checking if an auxiliary point in face coincides with the face boundary in 3D space. |
| 27890 | *Summary:* `BndLib_Add2dCurve::Add()` works incorrectly on some curves.<br><br>Draw Harness command `gbounding` has been extended to pass `curve2d` as arguments and to work in two modes: normal and optimal. |
| 27895 | *Summary:* Correction in the constructor `Extrema_ExtElC::Extrema_ExtElC (const gp_Lin&,const gp_Lin&,const Standard_Real)`<br><br>`Extrema_ExtElC` algorithm than finds minimum distance between 2 straight lines has been simplified. |
| 27930<br>27937 | *Summary:* XMT file conversion loops infinitely.<br><br>The intersection algorithm `IntWalk_IWalking` has been modified to estimate U- and V-ranges of future intersection curve(s) on the surface. This information is used in the stop-criterion of the algorithm instead of the full surface range as earlier. It allows reducing dependencies of the intersection result on the surface ranges. |
| 27946 | *Summary:* Out of range item access in offset algorithm.<br><br>Out of range access has been eliminated in method `BRepOffset_Inter2d::ConnexIntByInt`. |
| 27984 | *Summary:* Volume maker algorithm is unable to build all possible solids from the given faces<br><br>`BOPAlgo_PaveFiller::MakeBlocks()` now uses only the edges that belong to the faces when it checks two faces of a section for coincidence. |
| 27987 | *Summary:* `CellsBuilder` algorithm does not find shared common parts of the arguments<br><br>The possibility of common parts shared by the arguments is taken into account by method `BOPAlgo_CellsBuilder::IndexParts()`. |
| 27992 | *Summary:* `Extrema_ExtPS` crashes on a face without geometric surface.<br><br>The algorithm `BRepExtrema_DistShapeShape` has been protected against exceptions when non-geometric shape data is given on input, like a face containing only a triangulation or an edge containing only a polygon. Such faces/edges are ignored by the algorithm.<br><br>`BRepGProps::VolumeProperties()` now ignores faces without geometric surface to avoid access violation. |

| | |
|---|---|
| 28001 | *Summary:* Remove unused `BOPCol_VectorOfInteger.hxx`<br><br>The remains of `BOPCol_Array1` have been replaced with `BOPCol_NCVector`. |
| 28002 | *Summary:* Invalid result of Boolean Fuse operation<br><br>The initialization of the tree of bounding boxes of edges and vertices of the solid has been improved in class `BRepClass3d_SolidExplorer` to treat the internal/external parts of the solid correctly. |
| 28012 | *Summary:* Exception while intersecting two surfaces<br><br>The algorithm `IntPatch_PrmPrmIntersection` providing intersection of two surfaces now takes into account the case when they intersect only in one point. |
| 28028 | *Summary:* Solid becomes invalid after scaling.<br><br>The method `BRepTools_TrsfModification::NewCurve2d` uses `Precision::Confusion()` as a tolerance for `GeomLib::SameRange`. The tolerances of vertices are not taken into account while calculating tolerance of the transformed PCurve for the edge. |
| 28030 | *Summary:* Algorithm `GeomLib_CheckCurveOnSurface` takes too much time for Bspline curves with big number of knots.<br><br>The number of knots is now checked in `GeomLib_CheckCurveOnSurface` class. If there are too many knots the algorithm does not search for the distance between curves for each knot interval, but uses the predefined sample point distribution. |
| 28094 | *Summary:* Shape obtained after DRAW command `splitshape` has unnecessary high tolerance.<br><br>`Project` function from `LocOpe_WiresOnShape` class now checks the gap between ends of the p-curves having a common vertex in the parametric space of the face and if the face lies on the periodic surface. |

## Mesh

| | |
|---|---|
| 26321 | *Summary:* Crash in `BRepMesh_FastDiscret::Add`.<br><br>The class `BRepMesh_FaceAttribute` distinguishes constructor and parameters initialization. |
| 27239 | *Summary:* Meshing algorithm creates wrong triangulation.<br><br>The method `BRepMesh_FastDiscret::update` avoids inserting consequent duplicating nodes in `PolygonOnTriangulation`. |
| 27362 | *Summary:* Meshing performance.<br><br>The following modifications have been introduced to improve meshing performance:<br>▪ In class `BRepMesh_FastDiscretFace`: planes have been excluded from the procedure of inserting internal points; declaration of the container `aNewVertices` has been localized in each method where it is needed; the logic of method `insertInternalVerticesOther` has been corrected to separate the processes of removing extra points and addition of new points in different cycles, thus clearing the code; useful output of intermediate mesh has been inserted to a file in `control()` method for debug purposes (with definition `DEBUG_MESH`).<br>▪ Global functions `MeshTest_DrawTriangles` and `MeshTest_DrawLinks` draw mesh data in debug session.<br>▪ In method `BRepMesh_FastDiscret::Add` deflection calculations have been simplified for non-relative mode.<br>▪ The attribute `MinDist` has been replaced with `Deflection` in `EdgeAttributes` structure.<br>▪ The method `BRepMesh_Delaun::addTriangle()` has been protected against exception when an added triangle creates a third connection of a mesh edge.<br>▪ `Geom2dAdaptor_Curve` is used in `BRepMesh_EdgeTessellator` and `BRepMesh_EdgeTessellationExtractor` to provide b-spline cache while computing value on a curve<br>▪ Creation of a new b-spline is avoided in `BndLib_Box2dCurve::PerformBSpline` if the requested parameter range does not differ from natural bounds significantly<br>▪ In `GeomAdaptor` classes cache building is postponed till its actual usage. So, creation of an adapter to compute intervals of continuity does not lead to creation of internal cache.<br>▪ Transformed() function is not called in methods `Bezier` and `BSpline` from class `BRepAdaptor_Curve` if the transformation is identity.<br>▪ In classes `Geom_BSplineCurve`, `Geom_BSplineSurface`, `Geom_BezierCurve`, `Geom_BezierSurface`, `Geom2d_BSplineCurve` and `Geom2d_BezierCurve`, the method `Pole()` returns the point by `const` reference.<br>▪ In `CPnts_AbscissaPoint.cxx`, the derivative is computed by D1 instead of DN to make use of b-spline cache. |
| 27384<br>27416 | *Summary:* `BRepMesh_IncrementalMesh` does not take angular deflection into account for spun/elementary surfaces.<br><br>The deviation of normals at vertices of triangles is now checked for complex surface types different from Bezier and BSpline. Intermediate parameters greater than N-3 are not removed to have at least one parameter related to surface internals.<br><br>The angular deflection angle is checked before removal of intermediate parameters. |

| | |
|---|---|
| 27442 | *Summary:* Rotation sweep cannot be rendered in 3D.<br><br>The method `BRepMesh_FastDiscretFace::insertInternalVerticesCylinder` has been modified to avoid inserting internal nodes for Cylinder if it is less than `DefFace` value or in case of a long cylinder with a small radius due to protection against overflow during casting to integer. |
| 27490 | *Summary:* `BRepMesh`: Reduce number of memory allocations.<br><br>The number of memory allocations made by `BRepMesh_IncrementalMesh` algorithm has been reduced by grouping requests to larger blocks. Containers of types `sequence`, `list` and `map` are initialized with an instance of `NCollection_IncAllocator` to avoid occupying a huge amount of memory. Several arrays with the same and short life time are allocated in a single buffer array of necessary size.<br><br>Other changes:<br>▪ The function `filterParameters` from class `BRepMesh_FastDiscretFace` avoids excess memory allocations.<br>▪ The method `NCollection_CellFilter::Reset` accepts array by reference rather than by value.<br>▪ `Allocator()` method has been added in map, sequence and vector collection classes by analogy with list collection.<br>▪ Bounding box computation is avoided when no relative deflection is used<br>▪ Cycles by wires of face using `TopExp_Explorer` are converted to use `TopoDS_Iterator` instead.<br>▪ The method `BRepMesh_FastDiscret::Add` avoids storing sequences of faces and edges<br>▪ |
| 27595 | *Summary:* Faces without triangulations due to `gp_VectorWithNullMagnitude` exception.<br><br>Zero magnitude check has been added before `gp::Vec::Angle()` usage in method `BRepMesh_FastDiscretFace::insertInternalVerticesOther`. |
| 27959 | *Summary:* `BRepMesh_Delaun` produces mesh with gaps on internal edges.<br><br>New function `insertInternalEdges` from `BRepMesh_Delaun` class checks both sides of internal link for adjusted triangle and processes left and right polygons separately |
| 28118 | *Summary:* Mesh generation hangs then crashes<br><br>`BRepMesh_EdgeTessellator` algorithm has been protected against crash. |

## Shape Healing

| | |
|---|---|
| 25623 | *Summary:* Some tests become worse after approximation of p-curve by 2D line.<br><br>The tolerance of line/line analytical intersection has been fixed in method `IntCurve_IntConicConic::Perform`. |
| 26524 | *Summary:* ShapeFix_Face crash when performing fix on an invalid face.<br><br>It is now checked in method `ShapeFix_Face::Context()`if a non-null handle is returned. |

| | |
|---|---|
| 26786 | *Summary:* Segmentation violation exception raised if a shape to be fixed is null.<br><br>Check for null shape has been added in method `ShapeProcessAPI_ApplySequence::PrepareShape()`. |
| 26930 | *Summary:* `ShapeConstruct_ProjectCurveOnSurface` returns a B-Spline instead of line.<br><br>Method `ShapeConstruct_ProjectCurveOnSurface::PerformAdvanced` that checks closeness of 2dcurve to line during projection has been improved:<br>▪ For surfaces with C1 and above, the distance to the normal, not the distance to the surface is checked, for C0 surfaces the tolerance formula has been updated.<br>▪ Check for possible period jump in an internal point has been added.<br>▪ Cache saving for lines has been added; the function `fixPeriodicTroubles()` is updated using parameters from cache. |
| 27272 | *Summary:* `FixMissingSeam` function creates G1 seam curves.<br><br>Method `BRepLib::EncodeRegularity()`sets regularity `GeomAbs_CN` for edges lying on the same-domain surfaces (where derivatives on both surfaces are equal in all points).<br><br>New DRAW command `getedgeregularity` queries edge regularity on the specified faces. The command `edgeregul` has been removed because its functionality is provided by command `encoderegularity`. |
| 27464 | *Summary:* `BRepTools_ReShape` ends up with empty shapes.<br><br>`BRepTools_ReShape` has been protected against creation of empty shapes (e.g. wires having all edges removed). |
| 27541 | *Summary:* `ShapeFix_ComposeShell` allows usage of uninitialized value of the field `myInvertEdgeStatus`.<br><br>All fields of the class `ShapeFix_ComposeShell` are now initialized in the constructor. |
| 27729 | *Summary:* `UnifySameDomain`: allow the user to specify linear and angular tolerances.<br><br>The algorithm `ShapeUpgrade_UnifySameDomain` has been modified to consider linear and angular tolerances when checking if two faces are same domain:<br>▪ The tolerances can be set using new methods `SetLinearTolerance` and `SetAngularTolerance`.<br>▪ The draw command `unifysamedomain` accepts new parameters.<br>▪ The internal method `MergeSeq` avoids exception connected with access to unknown key in the data map. |
| 27781 | *Summary:* Exception in `ShapeFix_Shape` algorithm with option `FixSmallAreaWireMode`.<br><br>The method `ShapeAnalysis_Wire::CheckSmallArea()` now checks the area of the outer wire without hole-wires. The obsolete argument `theIsOuterWire` has been dropped. |
| 28143 | *Summary:* Location of the face is not taken into account in method `ShapeFix_Edge::FixAddPCurve`.<br><br>Method `ShapeFix_Edge::FixAddPCurve()` accepting face argument has been corrected to use the transformed surface for the projection tool, when the face is based on a surface with non-default location. |

## Visualization

| | |
|---|---|
| 00468<br>27818<br>27988<br>28061 | *Summary:* Provide an interface to define highlight presentation properties.<br><br>New wrapper `Graphic3d_HighlightStyle` allows setting up highlight mode presentation properties, such as highlight mode (box/color), color and transparency. It is now called by the API of all methods that use highlight or selection color. Correspondingly:<br><br>▪ Highlight in shading mode now supports transparency, implemented via blending;<br>▪ Transparency for selection can also be set, but implementing custom entity owners with additional presentation on application level;<br>▪ Deprecated methods `PrsMgr_PresentationManager::Highlight` and `PrsMgr_PresentationManager::BoundBox` that highlight object with a hard-coded color, as well as methods of `SelectMgr_EntityOwner`, that use presentation manager's highlight method have been removed;<br>▪ Methods of `IsHilighted AIS` context have been replaced with selection color checks;<br>▪ The API to store dynamic and selection highlight has been added to `Prs3d_Drawer` class;<br>▪ Customization of dynamic and selection highlight for particular objects is now available through `SelectMgr_SelectableObject::HilightAttributes()`;<br>▪ `AIS_InteractiveContext` highlight methods support individual highlight styles of interactive objects;<br>▪ New command `vselprops` allows customizing global selection and highlighting properties, such as auto-activation, pixel tolerance and colors. It replaces obsolete commands `vautoactivatesel` and `vselprecision`. |
| 21306<br>27680<br>27815<br>27883 | *Summary:* `AIS_InteractiveContext` – revise `DisplayedModes()` semantics.<br><br>The method `PrsMgr_PresentableObject::Presentations()` is the reliable way to query all display modes, for which the presentations have been computed (and possibly erased, i.e. invisible). The redundant method `AIS_InteractiveContext::DisplayedModes()` has been removed. The list of active display modes has been removed from `AIS_GlobalStatus`. |
| 22582 | *Summary:* Provide an API for dumping a sub-region of the viewport.<br><br>`Graphic3d_Camera` definition has been extended by optional `Tile` property `Graphic3d_CameraTile` (tile top-left corner, tile width and height, full frame width and height) with tile width and height equal to window size.<br><br>`V3d_View::ToPixMap()` now performs tiled dump when the image size exceeds hardware limits. |
| 23049<br>27763 | *Summary:* Remove `V3d_View::Print()` method.<br><br>`TKOpenGl` no longer depends on `FreeImagePlus` library for printing functionality on Windows platform. It still depends (optionally) on `FreeImage.dll`.<br><br>`Standard_DISABLE_DEPRECATION_WARNINGS` and `Standard_ENABLE_DEPRECATION_WARNINGS` have been added to eliminate compiler warnings about deprecated APIs within Draw Harness. |

| | |
|---|---|
| 23519<br>27596 | *Summary:* Mixing wireframe and shaded visualization impacts performance.<br><br>The function `computeFaceBoundaries()` from `StdPrs_ShadedShape` class does not create additional temporary buffer for edges.<br><br>`StdPrs_Isolines::AddOnTriangulation()` reconstructs polylines from segments projected onto triangulation.<br><br>`StdPrs_WFShape::Add()` packs isolines in a single group in presentation (instead of a per-face group) and groups lines with the same aspects.<br><br>The default width in methods `UIsoAspect()`, `VIsoAspect()` and `VIsoAspect()` from `Prs3d_Drawer` class has been changed from 0.5 to 1.0. |
| 24291<br>27947 | *Summary:* Move Z-buffer trihedron presentation from `TKOpenGl` to `TKV3d`.<br><br>The methods `TriedronDisplay()`, `TriedronErase()`, `ZbufferTriedronSetup()` and `TriedronEcho()` from `Graphic3d_Cview` class have been removed; a relevant functionality is provided by a new class `V3d_Trihedron`. |
| 24393 | *Summary:* Position objects with enhanced precision.<br><br>`PrsMgr_PresentableObject` and `Graphic3d_Structure` now consistently take and store `Handle(Geom_Transformation)` instead of `TColStd_Array2OfReal` and `Graphic3d_Mat4`.<br><br>Low-level advanced methods have been modified to pass `Handle(Geom_Transformation)`. High-level methods also accept the old syntax taking `gp_Trsf`. |
| 25180 | *Summary:* Homogeneous transformation API in `TKV3d`.<br><br>Public fields have been replaced by methods in class `Graphic3d_ZlayerSettings`.<br>New property `Origin` defines the local coordinate system for all Layer objects.<br>`Graphic3d_Cstructure` stores bounding box with double precision floats.<br>Frustum culling in `OpenGl_BVHTreeSelector` works with double precision floats.<br>The syntax of Draw Harness command `VZLayer` has been redesigned. |
| 25221 | *Summary:* Depth test errors in ray-tracing scene containing face outlines<br><br>New function `PolygonOffset` from `RaytraceBase.fs` calculates polygon offset for ray tracing. |
| 25576 | *Summary:* Implement `AIS_ConnectedInteractive::AcceptDisplayMode()`.<br><br>`AIS_ConnectedInteractive::AcceptDisplayMode()` has been implemented by redirecting to `myReference->AcceptDisplayMode()`. |
| 26434 | *Summary:* Textured objects should have priority over the environment mapping.<br><br>New parameter `UseEnvironmentTexture` has been added to `Graphic3d_ZlayerSettings`. The OSD layers do not use environment texture by default. Environment texture mapping can be enabled or disabled using `vzlayer` command. |

| | |
|---|---|
| 26434 | The following entities have been removed:<br>▪ Unnecessary files `Graphic3d_TypeOfSurfaceDetail.hxx` and `V3d_TypeOfSurface.hxx`.<br>▪ Functions `SurfaceDetailType` and `SetSurfaceDetailType` from `Graphic3d_Cview` and `OpenGl_View`.<br>▪ Functions `SurfaceDetailState` and `UpdateSurfaceDetailStateTo` from `OpenGl_ShaderManager`.<br>▪ Functions `SetSurfaceDetail` and `SurfaceDetail()` from `V3d_View`.<br>▪ Functions `SetDefaultSurfaceDetail` and `DefaultSurfaceDetail` from V3d_Viewer.<br>▪ Class `OpenGl_SurfaceDetailState`.<br>▪ Draw command `VsetTextureMode`. |
| 26512 | *Summary:* Build fails with VTK 6.2 and OpenGL2 rendering Backend.<br><br>Support of VTK with OpenGL2 Rendering Backend has been added. |
| 26641<br>27629<br>27728 | *Summary:* `TKOpenGl` – handle correctly transformation persistence within perspective projection.<br><br>Transformation persistence now properly supports perspective projection. The following new methods have been added:<br>▪ `AIS_InteractiveContext::SetTransformPersistence`, sets transformation persistence to object and selection.<br>▪ `SelectMgr_SelectionManager::UpdateSelection`, re-adds selectable object in BVHs in all viewer selectors.<br>▪ `SelectMgr_ViewerSelector::MoveSelectableObject`, moves object from a set of non-transformation persistence objects to a set of transformation persistence objects (or vice versa).<br>▪ `Graphic3d_TransformUtils::Convert` converts `gp_Trsf` to `Graphic3d_Mat4`.<br><br>The method `PrsMgr_PresentableObject::SetTransformPersistence()` has been removed. |
| 26809 | *Summary:* `TKOpenGl` – handle point arrays with per-vertex color within built-in GLSL programs.<br><br>Shader rendering of point sprites with per-vertex colors and shading have been fixed. Material properties now should be modified via `OpenGl_Context::SetShadingMaterial()` (instead of `OpenGl_ShaderManager`). |
| 26885<br>27717 | *Summary:* Drop redundant aspects from structure level.<br><br>The methods that allow defining aspects at structure level are generally unused as the aspects are usually defined at the level of the group.<br><br>Correspondingly, the structure aspect methods have been removed from `Graphic3d_Structure`, `Prs3d_Presentation`. `Graphic3d_Cstructure` and `OpenGl_Structure` classes. Undocumented test methods `DrawSphere()`, `SetPlane()` and `PickGrid()` from class `V3d` have been removed as well. |

| | |
|---|---|
| 26886<br>27953<br>27805 | *Summary:* TKV3d – eliminate global variables.<br><br>Global variables have been eliminated in TKV3d code to avoid issues in multi-thread applications (e.g. in case when the dedicated viewers are used from different threads). Consequently:<br><br>▪ New dummy class field has been created in `AIS_InteractiveContext` to have an empty `TopoDS_Shape` object.<br>▪ Static variables in classes `AIS_Point` and `PrsMgr_PresentableObject` have been renamed to local function variables.<br>▪ Unused static variables have been removed in classes `AIS_Shape` and `SelectMgr_SelectableObject`.<br>▪ Global static variables have become constant in classes `Graphic3d_MaterialAspect` and `V3d_Viewer`.<br>▪ Global variable `zRotation` from `V3d_View` has been moved to class field.<br>▪ Variable `theCurrentSelection` has become a field of `AIS_InteractiveContext` and `AIS_LocalContext` classes. Multiple selection is not used now, so each Context has its own selection.<br>▪ `myStructGenId` has been moved from `Graphic3d_StructureManager` to `Graphic3d_GraphicDriver` for identifying the structures in the driver. |
| 27008 | *Summary:* Polygonal selection algorithm does not work with `MeshVS_Mesh`.<br><br>Calculation of normals has been corrected in class `SelectMgr_TriangularFrustum`. |
| 27038<br>27591<br>28010 | *Summary:* Add new presentation for object manipulation.<br><br>New class `AIS_Manipulator` provides interactive services for manipulating with mouse local transformations of other interactive objects.<br><br>See more details in New Features section.<br><br>New function `DrawShaded` from `Prs3d_Arrow` builds a shaded (triangulated) arrow presentation. |
| 27202 | *Summary:* Add sensitivity for `Graphic3d_Buffer`.<br><br>New class `Select3D_SensitivePrimitiveArray` can be initialized directly from presentation data structures `Graphic3d_Buffer` defining triangulation or point set.<br><br>This class can also combine several elements into patches to reduce BVH initialization time at the expense of slower detection time.<br><br>▪ In `AIS_PointCloud::ComputeSelection()` selection is now computed on point set using `Select3D_SensitivePrimitiveArray` by default.<br>▪ In `PrsMgr_PresentableObject::Compute()`, redundant default argument value has been dropped.<br>▪ `BVH_Set` size is stored in the local variable to simplify debugging. |
| 27256 | *Summary:* Path Tracing - add support of alpha-channel of texture<br><br>Alpha-channel support has been added in ray-tracing and its processing in path-tracing has been improved to mix it with the native object transparency. |

| | |
|---|---|
| 27317 | *Summary:* Add protection against possible floating point overflows in BVH trees.<br><br>Missing implementation of `CenterOfGeometry` method has been added in `MeshVS_CommonSensitiveEntity`.<br><br>A check to prevent float overflow has been added to `OpenGl_BVHClipPrimitiveSet::Center`. |
| 27354 | *Summary:* `TKOpenGl` – add missing `OpenGl_TextureBufferArb` initializers for data in `Glushort` and `Glubyte` formats.<br><br>All variants of `Init` function from `OpenGl_VertexBuffer` are also supported by `OpenGl_TextureBufferArb`. Missing texture formats have been added to `OpenGl_GlFunctions.hxx`. |
| 27359 | *Summary:* Add support of flipping for textured text.<br><br>It is now possible to use previous model-view state for 3D text via `theHasOwnAnchor` option from `Prs3d_Text`. New functions `SetFlipping` and `HasFlipping` from `AIS_TextLabel` allow applying flipping. |
| 27360 | *Summary:* Remove obsolete anti-aliasing API.<br><br>Obsolete methods `SetAntialiasingOn()`, `SetAntialiasingOff()` and `Antialiasing()` have been removed from `V3d_View` class. |
| 27374<br>27375 | *Summary:* Optimize management of the scene bounding box.<br><br>`OpenGl_View` now caches bounding boxes per Z-layer (instead of bounding box of the entire scene in `Graphic3d_Cview`). Redundant invalidation of cached scene bounding box is now avoided when new presentation attributes are assigned to the graphic structure.<br><br>New methods `ConsiderZoomPersistenceObjects()` and `ConsiderZoomPersistenceObjects()`, which return zoom-scale factor, have been added in classes `Graphic3d_Cview`, `OpenGl_View` and `OpenGl_Layer`.<br><br>`ConsiderZoomPersistenceObjects()` function is called in method `V3d_View::FitMinMax`. |
| 27473<br>27834 | *Summary:* `SelectMgr_ViewerSelector` – iteration through detected Entities should be sorted.<br><br>The possibility to fetch the detected entity (usually topmost) for computing 3D coordinates under mouse cursor has been provided:<br><br>▪ `SelectMgr_SortCriterion` now stores detected `SensitiveEntity` and 3D point.<br>▪ The methods implementing class-as-iterator: `Init()`, `More()`, `Next()`, `Picked()`, `InitDetected()`, `MoreDetected()`, `NextDetected()` and `DetectedEntity()` from class `SelectMgr_ViewerSelector` have been replaced with `PickedData()`, `PickedEntity()` and `PickedPoint()` that access auxiliary information about the picked object in sorted order. Detection results should be assessed by using index. |

| | |
|---|---|
| 27477 | *Summary:* `Select3D_SensitiveCircle` always returns infinite depth value in boundary modes.<br><br>Unnecessary re-initialization of pick result has been removed from class `Select3D_SensitiveCircle`. |
| 27505 | *Summary:* `Font_FontMgr` – do not look for `fonts.dir` on OS X.<br><br>`Font_FontMgr::InitFontDataBase()` now uses the same approach on OS X and Android ignoring `fonts.dir` files, which are not used on these systems. |
| 27510 | *Summary:* `OpenGl_Window` – do not ignore backing store resize on OS X.<br><br>`OpenGl_Window::Resize()` now properly registers resize event when the window is moved to the screen with a different scale ratio. |
| 27523 | *Summary:* Selection owner contains obsolete shape.<br><br>The check for selection activation has been removed in `SelectMgr_SelectionManager::RecomputeSelection`. The given selection mode is now handled properly and single selection mode can be recomputed. |
| 27530 | *Summary:* `AIS_InteractiveContext::HilightNextDetected()` doesn't work in Neutral Point<br><br>The methods `HasNextDetected()` `HilightNextDetected()` and `HilightPreviousDetected()` from class `AIS_InteractiveContext` have been implemented for neutral point. |
| 27536 | *Summary:* Incorrect behavior of zoom persisted objects.<br><br>The behavior of `Graphic3d_TransformPers` in case of `Graphic3d_TMF_ZoomPers` type has changed. Zoom persistence mode now fixes the object in pixel coordinates and is independent on view size. |
| 27538 | *Summary:* `AIS_InteractiveContext::Display()` – do not erase previous display mode.<br><br>`AIS_InteractiveContext::Display()` does not erase old presentation modes of interactive object anymore. Instead, the method marks them hidden in the same way as `AIS_InteractiveContext::SetDisplayMode()` and `::UnsetDisplayMode()`. |
| 27549 | *Summary:* `OpenGl_View` – drop confusing misnamed methods `Width()` and `Height()`.<br><br>Methods `OpenGl_View::Width()` and `OpenGl_View::Height()` have been removed. |
| 27555 | *Summary:* `AIS_Shape` – own deviation coefficient change is not considered by Wireframe presentation mode.<br><br>New methods have been added in class `Prs3d_Drawer`:<br>▪ `UpdatePreviousDeviationCoefficient()` updates the previous value used for the chordal deviation coefficient to the current state.<br>▪ `UpdatePreviousDeviationAngle()` updates the previous deviation angle to the current value. |

| | |
|---|---|
| 27573 | *Summary:* `AIS_ColorScale::FindColor` does not take into account custom colors.<br><br>The method `AIS_ColorScale::FindColor()` now takes into account custom colors. |
| 27579 | *Summary:* `TKOpenGl` – skip FSAA flag within Path Tracing.<br><br>Ray-tracing now can work with enabled global illumination and antialiasing. |
| 27583 | *Summary:* `MeshVS` – handle `MeshVS_DA_SupressBackFaces` flag within `MeshVS_NodalColorPrsBuilder`.<br><br>`MeshVS_NodalColorPrsBuilder` and `MeshVS_ElementalColorPrsBuilder` now take into account `MeshVS_DA_SupressBackFaces` flag, which allows applying capping to the model. |
| 27590 | *Summary:* Ray Tracing – port to quad BVH trees (QBVH).<br><br>Binary BVH tree produced by building algorithms has been collapsed into 4-ary BVH (QBVH). The BVH traversal code in GLSL has been modified to process such trees correctly. This improves thread coherence, decreases BVH memory consumption (~2 times) and allows using a traversal stack of half size. |
| 27592 | *Summary:* `TKV3d`, `AIS_RubberBand` – SWIG error for method `ComputeSelection`.<br><br>Virtual & void modifiers of method `AIS_RubberBand::ComputeSelection` have been set in common order. |
| 27601 | *Summary:* `TKV3d`, `AIS_RubberBand` – polygonal selection mode invokes an exception with convex fill area.<br><br>The allocation of primitive array for triangles has been corrected in method `AIS_RubberBand::fillTriangles()`. |
| 27606 | *Summary:* View is blocking when MSAA has been overridden in graphics driver settings.<br><br>Method `OpenGl_View::blitBuffers()` tries disabling MSAA on `glBlitFramebuffer()` failure. |
| 27607 | *Summary:* Implement adaptive screen space sampling in path tracing.<br><br>Graphic3d_RenderingParams class has been extended with `AdaptiveScreenSampling` option (disabled by default). If this option is enabled, path tracing tries to adjust the number of samples for different screen areas. I.e. more complex areas (from the point of light conditions) are sampled more intensively, while simple areas are sampled very rarely. For example, caustics and glossy reflections are typical candidates for more precise sampling.<br><br>In general, this allows equalizing image convergence and avoids wasting resources for already converged areas. It is also possible to visualize sampling densities by enabling `ShowSamplingTiles` option (activating and deactivating this option does not affect the accumulated image).<br><br>Blending is now performed using OpenGL functionality, while ray-tracing shaders only output correct Z-value. |

| | |
|---|---|
| 27611 | *Summary:* Provide an interface to access selection frustum center points from `SelectBasics` level.<br><br>Getters for rectangular frustum center points have been added to `SelectBasics_SelectingVolumeManager`. |
| 27612 | *Summary:* `AIS_InteractiveContext` – handle `SelectMgr_EntityOwner::IsForcedHilight()` flag in neutral point.<br><br>The check for `IsForcedHilight` has been added to method `AIS_InteractiveContext::MoveTo`. |
| 27617 | *Summary:* `TKOpenGl` – apply highlighting color without disabling lighting.<br><br>`OpenGl_PrimitiveArray` now keeps lighting enabled when highlighting color is applied. |
| 27621 | *Summary:* Highlighting with color remains visible when shape is erased.<br><br>Method `AIS_InteractiveContext::EraseGlobal` has been fixed to hide color highlighting remains from the view when the shape is erased. |
| 27632 | *Summary:* `XCAFPrs_AISObject` – provide method for assigning new Label in the document.<br><br>It has become possible to change label in `XCAFPrs_AISObject` without creating a new presentation object. |
| 27633 | *Summary:* `TKOpenGl` – point sprites are inconsistent within Core and Compatible Profiles.<br><br>`OpenGl_ShaderManager::pointSpriteAlphaSrc()` now does not return alpha from Red channel for RGBA marker texture in Core profile.<br>The code for drawing points without texture has been restored in method `OpenGl_ShaderManager::prepareStdProgramFlat()`.<br><br>Built-in GLSL programs now flip `.y` in shaders instead of relying on `GL_POINT_SPRITE_COORD_ORIGIN` unavailable on OpenGL ES.<br><br>A new sample `markers.tcl` has been added for testing marker orientation. |
| 27655 | *Summary:* `AIS_Triangulation` disappears after setting non-zero transparency.<br><br>The following methods have been added:<br>▪ `AIS_Triangulation::SetTransparency()` sets `aValue` for transparency in the reconstructed compound shape.<br>▪ `AIS_Triangulation::UnsetTransparency` removes the setting for transparency in the reconstructed compound shape<br>▪ `AIS_Triangulation::HasVertexColor` returns true if the triangulation has vertex colors.<br>▪ `Graphic3d_ArrayOfPrimitives::AddVertex()` adds a vertex and its color in the vertex array and returns the actual vertex number.<br>▪ `Graphic3d_ArrayOfPrimitives::SetVertexColor()` changes the vertex color of rank `<theIndex>` in the array.<br><br>The type of color of array in `AIS_Triangulation` is now `Graphic3d_Vec4ub`. |

| | |
|---|---|
| 27668 | *Summary:* TKD3Dhost – fix accessing function list within Core Profile.<br><br>Broken code has been fixed in class D3Dhost_FrameBuffer. |
| 27670 | *Summary:* Avoid duplication of structures defining primitive array presentation aspects.<br><br>The definition of primitive array presentation aspects has been improved.<br><br>The following modifications have been implemented to improve the definition of primitive array presentation aspects:<br><br>▪ New convenient structure Quantity_ColorRGBA holds Vec4 for OpenGL.<br>▪ Graphic3d_PolygonOffset replaces TEL_POFFSET_PARAM.<br>▪ Aspect_HatchStyle enumeration now follows values of TEL_HS_*** for compatibility. Duplicating definition of Hatch Styles TEL_HS_*** and TelCullMode enumeration have been removed.<br><br>▪ Aspect_AspectLine has been merged into Graphic3d_AspectLine3d.<br>▪ Aspect_AspectMarker has been merged into Graphic3d_AspectMarker3d.<br>▪ Aspect_AspectFillArea has been merged into Graphic3d_AspectFillArea3d.<br>▪ Graphic3d_CaspectFillArea has been removed.<br><br>▪ OpenGl_AspectLine now stores Graphic3d_AspectLine3d as class field.<br>▪ OpenGl_AspectMarker now stores Graphic3d_AspectMarker3d as class field.<br>▪ OpenGl_AspectText now stores Graphic3d_AspectText3d as class field.<br>▪ OpenGl_AspectFace now stores Graphic3d_AspectFillArea3d as class field.<br><br>▪ Back face culling is now enabled in Graphic3d_AspectFillArea3d by default.<br>▪ TKOpenGl now relies on Graphic3d_Group::IsClosed() flag to disable face culling.<br>▪ StdPrs_ShadedShape does not modify aspect for different culling modes.<br>▪ Obsolete headers InterfaceGraphic_Graphic3d.hxx, InterfaceGraphic_telem.hxx and InterfaceGraphic_tgl_all.hxx have been removed.<br>▪ Drawing a non-indexed array from VBO has been fixed in OpenGl_PrimitiveArray::drawEdges()<br>▪ Missing initialization of 3D arrow aspect has been added in AIS_Dimension::DrawArrow()<br>▪ AIS_Manipulator::Compute()creates a dedicated Face Aspect for each axis.<br>▪ V3d_CircularGrid and V3d_RectangularGrid now create dedicated line aspects with different color.<br>▪ AIS_InteractiveObject::SetMaterial() does not modify the global ShadingAspect.<br><br>In samples:<br><br>▪ Broken custom presentations Sample2D_Text and ISession_Text have been replaced by AIS_TextLabel.<br>▪ Group definition without presentation aspects has been fixed in ISession2D_Curve::Compute().<br>▪ Assigning custom attributes has been fixed in GeomSources. |

| | |
|---|---|
| 27682 | *Summary:* Provide method `Prs3d_Drawer::SetShaderProgram()` for setting program.<br><br>New public method `AIS_ColoredShape::CustomAspectsMap()` allows accessing aspects map.<br><br>New method `AIS_InteractiveObject::SynchronizeAspects()` synchronizes all primitive aspects at low-level (`TKOpenGl`) after their modification.<br><br>The class `AIS_ColoredDrawer` has been moved to the dedicated file. |
| 27688 | *Summary:* `AIS_Dimension` - add possibility to set custom text value<br><br>New function `SetCustomValue` from `AIS_Dimension` shows a real value or the name of its parameter in dimension presentation for the value. |
| 27692 | *Summary:* `AIS_AngleDimension` - exterior angle and arrows visibility improvements.<br><br>New Draw command `vdimangleparam` allows:<br><br>▪ visualizing either the exterior or the interior angle in a presentation. Previously the minimal angle was computed and visualized.<br>▪ hiding the first or the second arrow in the presentation to show the direction of angle: from one line to another. |
| 27715<br>28072<br>28062 | *Summary:* Turn off deprecated OpenGL fixed-function pipeline by default.<br><br>The flag `ffpEnable` is now initialized in class `OpenGl_Caps` to false on desktop OpenGL.<br>The method `OpenGl_Context::init()` now prints error if OpenGL version does not support GLSL and switches `OpenGl_Caps::ffpEnable` to true.<br><br>Deprecated vector export through `gl2ps` and unsupported shading models have been removed from MFC sample. |
| 27724 | *Summary:* `TKV3d` – Null handle check missing in `AIS_InteractiveContext::SelectedShape()`.<br><br>Null handle check has been added in method `AIS_InteractiveContext::SelectedShape()`. |
| 27727 | *Summary:* `Graphic3d_ArrayOfPrimitives::SetVertexColor()` – define opaque color for RGB input.<br><br>RGBA color has become opaque in `Graphic3d_ArrayOfPrimitives::SetVertexColor()`. |
| 27731 | *Summary:* `OpenGl_AspectMarker` – handle fractional marker Scale when sharing resources.<br><br>`OpenGl_AspectMarker::Resources::SpriteKeys()` now encodes the decimal number of marker scale into resource key, since built-in markers are defined with 0.5 scale step. |

| | |
|---|---|
| 27735 | *Summary:* `OpenGl_ShaderManager` – fix clipping state management.<br><br>`OpenGl_CappingAlgo::RenderCapping()` now updates the clipping state in Shader Manager. `OpenGl_View::render()` clipping state setup has been moved to `OpenGl_View::renderScene()`.<br><br>`OpenGl_ShaderManager` now converts position to homogeneous coordinates within GLSL clipping code. This fixes capping plane rendering with clipping planes applied.<br>Possible Clipping planes misconfiguration when FFP is used has been fixed. The error-prone method `OpenGl_Clipping::AddWorldLazy()` has been dropped. |
| 27739 | *Summary:* `TKV3d` – implement individual acceleration data structure for selection of 2D persistent objects.<br><br>Low-level selection algorithms now explicitly support 2D transformation persistent objects:<br>▪ The lists of objects in `SelectMgr_SelectableObjectSet` have been split onto three subsets: regular, 3d-persistent and 2d-persistent. Each subset has an individual `BVHtree`. The algorithms update only the trees that are really required for the actual camera state.<br>▪ `SelectMgr_ViewerSelector` explicitly supports Eye space selection operations on BVH tree for 2d-persistent subset. The change of camera position does not anymore affect the acceleration data structure (BVH tree) of 2d-persistent selectable objects. |
| 27750 | *Summary:* `V3d_View` – remove unused `Zclipping` and `Zcueing` functionality.<br><br>The methods implementing Z-clipping planes: `SetZClippingType()`, `SetZClippingDepth()`, `SetZClippingWidth()`, `SetZCueingDepth()`, `SetZCueingWidth()`, `SetZCueingOn()` and `SetZCueingOff()` have been removed from class `V3d_View`.<br><br>Any applications that have used them should define a Clipping Plane via the general mechanism (e.g. `V3d_View::AddClipPlane()`) and update its position on camera movements. |
| 27751 | *Summary:* `Graphic3d_ClipPlane` – add option to inherit material from object.<br><br>Capping planes now can inherit a material from the object itself. This mode is turned on by the method `OpenGl_CappingAlgo::RenderCapping()`.<br><br>`MeshVS_ElementalColorPrsBuilder::Build()` now creates a single primitives group for triangles, thus `Closed` flag is correctly applied. |
| 27751 | Redundant methods `DrawGroups()` and `renderClosedGeometry()` have been removed from class `OpenGl_Structure`.<br><br>Draw Harness command `vaspects` now preserves display mode and location when assigning sub-shape aspects.<br>The syntax of command `vclipplane` has been revised:<br>▪ Redundant arguments `change`, `view` and `object` have been eliminated.<br>▪ Multiple parameters can be passed within a single call.<br>▪ The new plane is created implicitly with `create` command.<br>▪ The argument `maxplanes` does not require view name.<br>▪ The argument `delete` does not throw TCL exception for a non-existing plane.<br>▪ The argument `view` used without list now applies to the active view.<br>▪ `Handle *` and `ALL` within `delete` remove all defined planes. |

| | |
|---|---|
| 27755 | *Summary:* V3d_View::ToPixMap() – fix image dump with MSAA turned on.<br><br>OpenGl_View::Redraw() now handles correctly myTransientDrawToFront with MSAA turned on and resolution not equal to window size. |
| 27756 | *Summary:* Add Draw() method taking Graphic3d_Group to tools Prs3d_Arrow and Prs3d_Text.<br><br>New Draw() methods have been added to tools Prs3d_Arrow and Prs3d_Text to take Graphic3d_Group directly. The old Draw() methods from Prs3d_Presentation have been deprecated. |
| 27757 | *Summary:* Handle child objects in selection manager regardless of HasOwnPresentations() flag.<br><br>All methods of SelectMgr_SelectionManager now process children first, then check HasOwnPresentations() flag. |
| 27764<br>28056 | *Summary:* Add functionality for animation of 3D camera and interactive objects.<br><br>New classes AIS_Animation, AIS_AnimationCamera and AIS_AnimationObject have been added for animation of 3D camera and interactive objects.<br><br>Correspondingly:<br>▪ Draw Harness command vanimation has been modified to manage animation timeline.<br>▪ Command vfit has been extended with option -noupdate.<br>▪ Formatting of vviewparams command output has been improved.<br>▪ Functionality of commands vlocreset, vlocmove, vloctranslate, vlocrotate, vlocmirror and vlocscale has been merged into vlocation/vsetlocation.<br>▪ vlocation now can print the current local transformation of the object.<br>▪ v3d/ivtk test group does not call vfit anymore. |
| 27777<br>27916 | *Summary:* TKV3d - Standard interactive objects clear the presentation in Compute().<br><br>NULL shape check has been added in AIS_ColoredShape::Compute().<br><br>Redundant clearance has been removed in Compute() methods from AIS package classes, as the presentation is always cleared in advance within PrsMgr_PresentationManager::Update(). |
| 27783 | *Summary:* XCAFPrs_AISObject – override method SetMaterial().<br><br>XCAFPrs_AISObject::SetMaterial() now changes the default material of the object but preserves custom XDE styles to properly handle material updates.<br><br>▪ XCAFPrs_AISObject::Compute() does not reset the map of custom aspects at each call, to keep in sync aspects across multiple presentations.<br>▪ Private methods and class fields in AIS_InteractiveContext have become protected to allow inheritance. |

| | |
|---|---|
| 27787 | *Summary:* TKOpenGl – Optimize rendering by additional check whether the object is fully clipped or not.<br><br>OpenGl_Structure::Render() now checks if the structure is entirely clipped to skip rendering at all, or entirely NOT clipped to disable clipping / capping plane.<br><br>OpenGl_ShaderManager now defines dedicated GLSL programs for one and two clipping planes to optimize rendering with slow hardware. |
| 27789 | *Summary:* TKOpenGl – improve compatibility with new OpenGL ES devices.<br><br>The definition of global constants in optimized Anaglyph shader has been fixed in method OpenGl_ShaderManager::prepareStdProgramStereo().<br><br>OpenGl_Context now loads GL_KHR_debug extension within OpenGL ES context. |
| 27793<br>27817<br>28093 | *Summary:* Object drifts at zoom within Graphic3d_TMF_TriedronPers applied.<br><br>Graphic3d_TransformPers now takes Graphic3d_Camera definition as argument for methods applying transformation.<br><br>Graphic3d_TransformPers::Apply() computes Graphic3d_TMF_TriedronPers transformation in the following way:<br>▪ The object is moved onto Z focus distance.<br>▪ The object is expected to be defined in pixels.<br>▪ The Z coordinate on anchor point is used as offset from the view corner in pixels.<br>▪ It is possible to define not only the view corners, but also the midpoint of the side.<br><br>Graphic3d_TMF_TriedronPers now works with perspective projection.<br>OpenGl_Layer::BoundingBox() now takes into account the bounding box of Graphic3d_TMF_TriedronPers presentations for Z-fit operation. |
| 27796 | *Summary:* Allow 3D objects with Graphic3d_TMF_2d flag.<br><br>Usage of 3D objects with Graphic3d_TMF_2d flag has been enabled:<br>▪ Graphic3d_TransformPers::Apply() now does not reset projection matrix for objects with Graphic3d_TMF_2d flag.<br>▪ Useless flag Graphic3d_TMF_2d_IsTopDown has been removed.<br>▪ SelectMgr_SelectableObjectTrsfPersSet does not skip Graphic3d_TMF_2d presentations.<br>▪ OpenGl_Layer::BoundingBox() takes into account Graphic3d_TMF_2d presentations for proper Z-fit.<br>▪ AIS_ColorScale now uses “lazy” mode for rendering labels (considering 2D persistence to be already defined within the entire structure).<br>▪ OpenGl_Layer::updateBVH() now updates myAlwaysRenderedMap to handle dynamic transformation persistence flag change without redisplaying the object. |
| 27797 | *Summary:* Consider Zlayer properties while sorting list of picked entities.<br><br>OpenGl_GraphicDriver::Zlayers() and V3d_Viewer::GetAllZLayers() now return the layers sequence following rendering order (taking into account IsImmediate flag).<br><br>StdSelect_ViewerSelector3d::Pick() now sorts results taking into account Zlayers flags. |

| | |
|---|---|
| 27811 | *Summary:* Allow building `TKOpenGl` with OpenGL ES on Windows.<br><br>The method `OpenGl_ShaderManager::prepareStdProgramFboBlit()` tries using the extension `GL_EXT_frag_depth` within OpenGL ES 2.0 when OpenGL ES 3.0 is not available.<br><br>`OpenGl_View::blitBuffers()` disables `Depth` test when copying depth values is not supported by OpenGL ES 2.0 hardware.<br>`OSD_Environment` defines the global environment map for emulating desktop behavior on UWP. |
| 27813 | *Summary:* Add method `V3d_View::DiagnosticInformation()` similar to `vglinfo` command.<br><br>New method `V3d_View::DiagnosticInformation()` provides access to low-level OpenGL context information for automated diagnostic reports or displaying About System in the application. |
| 27816<br>27944<br>27945 | *Summary:* Provide an API for overriding clipping planes list.<br><br>The API of clipping planes has been extended in the following way:<br><br>▪ `Graphic3d_SequenceOfHClipPlane` now inherits `Standard_Transient`.<br>▪ `PrsMgr_PresentableObject`, `Graphic3d_Structure`, `Graphic3d_Cstructure` `V3d_View` and `OpenGl_View` now manage the plane list by `Handle`.<br>▪ The getters `::GetClipPlanes()` have been removed,<br>▪ The setters taking non-handle `::SetClipPlanes()` have been marked deprecated.<br>▪ `OpenGl_Structure::Render()` and `SelectMgr_ViewerSelector::checkOverlap()` now disable global (view) clipping planes for objects with flags `Graphic3d_TMF_TriedronPers` and `Graphic3d_TMF_2d` or with new flag `Graphic3d_SequenceOfHClipPlane::ToOverrideGlobal()`.<br>▪ `OpenGl_Clipping` now implements the interface for managing clipping planes without copying the sequences. The filtering of duplicates is no more performed by `OpenGl_Clipping` application is responsible to not do this. `OpenGl_Clipping` tries avoiding unnecessary allocations for managing the list of active planes.<br>▪ MFC sample now uses `V3d_View::ClipPlanes()` method.<br>▪ `SelectMgr_ViewerSelector::checkOverlap()` and `OpenGl_Structure::Render()` clip the entire zoom/rotate persistence object by checking the anchor point with global clipping planes. |
| 27819 | *Summary:* Provide the possibility to redefine `SelectMgr_SelectableObject::UpdateSelection`.<br><br>It has become possible to redefine `SelectMgr_SelectableObject::UpdateSelection`. |
| 27836 | *Summary:* `TKOpenGl` – gradient background is lost at some camera positions.<br><br>`OpenGl_View::ZlayerBoundingBox()` now adds screen background plane for proper Z-fit.<br><br>In method `Graphic3d_TransformPers::Compute()`, projection matrix has been excluded from `math` to eliminate floating point computation error. |

| | |
|---|---|
| 27853 | *Summary:* `OpenGl_Texture` – optimize sequential upload of texture image.<br><br>New methods `Revision()` and `UpdateRevision()` from class `Graphic3d_TextureRoot` mark updates in texture data source.<br><br>`OpenGl_AspectFace` handles `Graphic3d_TextureRoot::Revision()` changes. `OpenGl_Texture::Init()` now patches already allocated texture image when possible.<br><br>Methods `HasMipMaps()` and `SetMipMaps()` from class `Graphic3d_Texture2D` configure `MipMap` usage (as an alternative to sub-classing). |
| 27860 | *Summary:* Clean up Transformation Persistence API.<br><br>The following improvements have been introduced in Transformation Persistence API:<br><br>▪ Redundant `NCollection_Handle` usage in method `Graphic3d_Camera::TransformMatrices` has been replaced with validity flags.<br>▪ `Graphic3d_TransModeFlags` is now defined as enumeration, instead of integer bit flags. `Graphic3d_TMF_PanPers` and `Graphic3d_TMF_FullPers` have been removed.<br>▪ `Graphic3d_TMF_ZoomRotatePers` has been extended to define independent offset in pixels within X and Y coordinate.<br>▪ `Graphic3d_TransformPers` defines dedicated constructors for 3D persistence (zoom / rotate) and 2D persistence (2d / trihedron). 2D persistence now supports dedicated values for X and Y offsets.<br>▪ The corner is now specified by enumeration `Aspect_TypeOfTriedronPosition` instead of indirect interpretation of anchor point values.<br>▪ `PrsMgr_PresentableObject` and `Graphic3d_Cstructure` hold `Handle(Graphic3d_TransformPers)` instead of a value.<br>▪ Methods `SetTransformPersistence` and `TransformPersistence()` work with `Handle(Graphic3d_TransformPers)`. |
| 27867 | *Summary:* `AIS_InteractiveContext::HasSelectedShape()` returns true only if the selected interactive is an instance of `AIS_Shape`.<br><br>`AIS_InteractiveContext::HasSelectedShape()` now checks `SelectedOwner` type and returns true if the owner is an instance of `StdSelect_BRepOwner`. |
| 27868 | *Summary:* `AIS_InteractiveContext::Remove()` might keep removed object in selection list.<br><br>Method `AIS_InteractiveContext::unhighlightOwners` has been corrected to handle all objects in the selection regardless of owner's selection state. |
| 27893 | *Summary:* `AIS_InteractiveContext::SetSelected` does not work.<br><br>The method `AIS_InteractiveContext::SetSelected` checking global selection existence has been corrected.<br><br>The remaining occurrences of hard-coded 0-selection mode have been corrected to use the global selection mode; |

| | |
|---|---|
| 27899 | *Summary:* Ray Tracing – Provide ability to rebuild GLSL shaders on the fly.<br><br>It is now possible to rebuild ray tracing shaders on the fly using `vrenderparams -rebuild` command. This allows analyzing the impact of different shader modifications without re-launching and re-configuring the scene. |
| 27943 | *Summary:* Fix broken shading by positional light for object with local transformation.<br><br>In `OpenGl_ShaderManager` model-world matrix is not applied to the light sources, which are expected to be defined in world-space. |
| 27952 | *Summary:* Do not reset `RayTracing` state on changing structures not in main Z-Layer.<br><br>Ray Tracing state is not reset when the ray-traceable structures that are not present in the main Z-Layer are removed from `OpenGl_LayerList`. |
| 27957 | *Summary:* `AIS_InteractiveContext` – protect from displaying the same `AIS_InteractiveObject` within multiple contexts.<br><br>`AIS_InteractiveContext` methods that add object to the context now throw `Standard_ProgramError` exception if the object has been already displayed in another context.<br><br>`AIS_InteractiveContext::Remove()` nullifies the context assigned to the object. It inherits from `Standard_Transient` instead of deprecated `MMgt_Tshared` and defines C++ destructor instead of method `Delete()`.<br><br>Undocumented property `State()` has been removed from `AIS_InteractiveObject`. Undocumented property `Users()` has been moved to `AIS_IdenticRelation`.<br><br>Draw Harness command `vclose` now clears `AIS_InteractiveContext` content before nullifying it to ensure that objects have been properly removed.<br><br>`AIS_MultipleConnectedInteractive` now overrides method `SetContext()` to assign context for children objects. |
| 27961 | *Summary:* Remove unused and non-working `OpenGl_AVIWriter`.<br><br>Outdated class `OpenGl_AVIWriter` has been removed. |
| 27969 | *Summary:* Add interfaces to access selecting volumes from `SelectMgr_SelectingVolumeManager`.<br><br>An interface to access the current selecting volume for complex objects that are calculated during rendering has been provided. The following features have been added:<br>▪ Getter for selection frustum computed during last run of selection mechanism in `SelectMgr_SelectingVolumeManager`;<br>▪ Getters for frustum planes in `SelectMgr_BaseFrustum` and its successors;<br>▪ API for applications to get frustum planes in `SelectBasics_SelectingVolumeManager`. |
| 27972 | *Summary:* Remove unused and not implemented property `V3d_View::EnableGLLight()`.<br><br>Unused function `V3d_View::EnableGLLight()` has been removed. |

| | |
|---|---|
| 27974 | *Summary:* Ray tracing – Improve ray tracing engine.<br><br>The following features have been introduced to improve ray tracing:<br>▪ The problem with high level of noise caused by glossy objects resulting in significant convergence time has been resolved by using Multiple Importance Sampling (MIS) strategy, which mixes contributions from both explicit and implicit light paths.<br>▪ Sampling and handling subroutines have been reviewed to better handle several light sources.<br>▪ Light distance is taken into account in light source intersection.<br>▪ New TCL sample – OCCT Ball model demonstrates physically-based materials.<br>▪ Potential "Error: Failed to upload light source buffer" has been fixed on NVIDIA GPUs. |
| 27986 | *Summary:* `AIS_InteractiveContext::SetLocation()` does not update dynamic highlighting.<br><br>`AIS_InteractiveContext::SetLocation()` updates `HilightMode` instead of `DisplayMode`. Accessing NULL presentation has been fixed in `PrsMgr_PresentationManager: :UpdateHighlightTrsf()`. |
| 28031 | *Summary:* TKV3d – do not use height/width ratio of window during `V3d_View::FitAll()`.<br><br>The method `V3d_View::FitAll()` now uses the aspect ratio from camera rather than from virtual window |
| 28035 | *Summary:* `V3d_Trihedron::compute()` endlessly creates new graphic groups.<br><br>`V3d_Trihedron::compute()` now reuses existing groups in the structure and resets the flag `myToCompute`. |
| 28036 | *Summary:* `AIS_ColoredShape` – handle correctly nested compounds within Shaded display mode<br><br>`AIS_ColoredShape::Compute()` now parses nested compounds in two passes to handle complex cases with compounds used for grouping styles. |
| 28099 | *Summary:* `OpenGl_Text` - handle `DIMENSION` and `SUBTITLE` styles within Core Profile.<br><br>`OpenGl_Text` now creates VBO for drawing background rectangle. |
| 28101 | *Summary:* `Select3D_SensitiveSet` - fix `NULL` dereference on re-adding the same sensitivity.<br><br>`Select3D_SensitiveSet` now stores `BVH_PrimitiveSet` as class field (no dynamic allocation) and `BVH_PrimitiveSet` subclass now stores raw pointer to `Select3D_SensitiveSet` (no smart pointer).<br><br>`Select3D_BVHPrimitiveContent` definition has been moved into Select3D_SensitiveSet class definition to avoid confusion. |
| 28114 | *Summary:* Ray tracing - Make ray tracing mode interactive in high resolutions.<br><br>The number of bounces in HD screen resolutions is now dynamically adjusted depending on the type of ray-surface interaction and gradually increased when the camera is stabilized. |

| | |
|---|---|
| 28127 | *Summary:* Transparent object breaks Z-Layer depth buffer clearing.<br><br>`OpenGl_LayerList::Render()` now calls `glDepthMask(GL_TRUE)` before clearing depth buffer.<br>Depth buffer is now cleared even if a `ZLayer` with this command has no structures. |

## VIS

| | |
|---|---|
| 27567<br>27734 | *Summary:* Possible memory leaks due to use of plain pointers.<br><br>Sources of memory leaks have been eliminated in `IVtkDraw`, `IvtkTools` and `IVtkOCC` packages. |
| 27671 | *Summary:* Basic `Ivtk` tools do not allow easy sub-classing.<br><br>The following modifications have been made to facilitate the integration of `Ivtk` functionalities into custom applications:<br><br>▪ Missing `Standard_EXPORT` macro has been added in `IVtkOCC_Shape.hxx`.<br>▪ Members of `IvtkTools_ShapeDataSource` are declared as protected. |

## Data Exchange

| | |
|---|---|
| 27169 | *Summary:* Suspicious behavior of importing names during STEP import.<br><br>Method `Tcollection_AsciiString::UsefullLength()` is now used instead of `Tcollection_AsciiString::Length()` to avoid using empty strings in the assembly names in XCAF document after reading step file when description or name is defined by only a 1-space character method. |
| 27235 | *Summary:* Export GDT: Annotation plane and Presentation.<br><br>`Null_Style` STEP type has been implemented. Annotation planes and presentation are exported as tessellated geometry. |
| 27304 | *Summary:* Implementation of descriptions for Dimensions.<br><br>The possibility to save and import/export from STEP text descriptions for Dimensions has been implemented. |
| 27329 | *Summary:* Export to STEP failure.<br><br>`ProcessShape` is now available for non-manifold shapes. Function `mergeInfoForNM` from class `STEPControl_ActorWrite` binds already written shared faces to STEP entity. |
| 27336<br>28147 | *Summary:* Update STEP entities, according to AP242.<br><br>It is now possible to export STEP with header corresponding to AP242. |
| 27372 | *Summary:* Import/Export dimension text position.<br><br>The position of annotation plane can be imported from a STEP file not written in OCCT in the following way:<br>▪ If the annotation plane lies in the presentation bounding box, its position will be saved as the text position.<br>▪ If otherwise, the text position will be calculated using the center of bounding box.<br><br>The following Draw commands have been implemented:<br>▪ `addGDTPosition` and `getGDTPosition` to set and get the orientation and position of dimension text.<br>▪ `addGDTPresentation` and `getGDTPresentation` to set and get the presentation. |
| 27404 | *Summary:* Access violation when reading STEP file<br><br>Check on the null handle has been added in method `STEPControl_Reader::NbRootsForTransfer()`. |
| 27427 | *Summary:* Exception on loading a DE package to Draw.<br><br>The range of parameter is now checked in method `IFSelect_WorkSession::SetParams` to avoid exception. |
| 27447 | *Summary:* Add support for long IGES entity names.<br><br>The `IGESCAFControl_Writer` supports labels/names with more than 8 characters according to IGES 406 / 15. |

| | |
|---|---|
| 27455 | *Summary:* Implementation of connection points.<br><br>Import and export to STEP connection points have been implemented for dimensions. Connection points will be exported to STEP as Derived geometry. |
| 27532 | *Summary:* Errors on reading stp-file.<br><br>The warning about a non-alphabetical order of entities has been added in method `StepData_StepReaderData::SetRecord`. |
| 27570 | *Summary:* Unacceptable performance during reading large STEP files<br><br>The performance of assembly translation from STEP has been improved. |
| 27575 | *Summary:* Reading STEP file gives an empty result.<br><br>The following protections have been added in the algorithms for reading STEP files:<br>▪ against exception when reading a BSpline curve with zero control points;<br>▪ against exception due to null binder when writing an XDE document. |
| 27622 | *Summary:* STL file having less than 4 triangles cannot be read.<br><br>Minimum file size check has been dropped in method `RWStl::ReadBinary()`. |
| 27645 | *Summary:* Access violation when reading STEP AP242 file.<br><br>Protection against invalid input has been added in `STEPCAFControl_Reader`. |
| 27721 | *Summary:* Add the possibility to read STEP file with invalid `shape_representation` entity.<br><br>New parameter `step.all.shapes` allows reading all top level (not shared by any other entities) solids (`manifold_solid_brep`) and shells (`shell_based_surface_model`) with `ReadStep` command. |
| 27807 | *Summary:* Units are lost during import presentation of GDT.<br><br>The import of `draughting_model` subtype has been implemented. |
| 27808 | *Summary:* Some `geometric_tolerances` are not imported.<br><br>The entity `datum_feature` is processed when `shape_aspects` of GDT are collected. |
| 27934 | *Summary:* Implement STEP common labels<br><br>Common labels have been implemented as a special dimension type containing only a presentation and an annotation plane without any links to shapes. |
| 27975 | *Summary:* Add the possibility to connect DGTs to vertices.<br><br>New function `FindShapeIndexForDGT` from `STEPCAFControl_Reader` finds shape index in the map of imported shapes.<br><br>New function `FindPDS` from `STEPCAFControl_Writer` finds `Product_definition_shape` entity for the given entity. |

## Draw

| | |
|---|---|
| 24932 | *Summary:* addsweep does not check if the supplied variable contains a shape.<br><br>Check for null shape has been added in Draw command addsweep. |
| 27245 | *Summary:* DRAW command to test binary persistence for shapes.<br><br>New static methods Read and Write from BinTools class and the corresponding Draw commands binsave and binrestore allow storing and retrieving a shape in binary format without loss of precision. |
| 27313 | *Summary:* Exception during WriteStep with PMI.<br><br>New Draw commands setDatumPosition and getDatumPosition allow setting/getting datum position. Datum position is checked during export of datum_system. |
| 27332 | *Summary:* Duplicate output from Draw Interpreter in cout in the debug function Draw_Eval.<br><br>The output is duplicated to see the result in terminal without using puts command. |
| 27624 | *Summary:* ViewerTest – vdrawtext command ignores text display options.<br><br>The command vdrawtext sets the specified text display type regardless of other options. |
| 27696 | *Summary:* Return max distance in Draw command xdistcs.<br><br>A message about maximal distance has been added for xdistcs command. |
| 27879 | *Summary:* ViewerTest – disable Vsync within non-interactive mode.<br><br>By default, Vsync is disabled in non-interactive mode to improve the performance. |
| 27905 | *Summary:* Randomly directed arrows on test v3d mesh B7.<br><br>Draw command meshvectors now shows the default vector (0,0,1) for volume entity. |
| 28095 | *Summary:* ViewerTest – use RGBA format instead of BGRA within vreadpixel<br><br>OpenGl_Workspace::BufferDump() now implicitly converts RGBA dump into requested BGR, BGRA and RGB image.<br><br>DRAW command dversion reports the OpenGL variant (desktop or ES). |

## Samples

| | |
|---|---|
| 25362 | *Summary:* Conversion to B-Spline doesn't show B-Spline surface on Windows 8.<br><br>Constants `WM_MOUSEFIRST` and `WM_PAINT` have been added in `WaitForInput` method for correct handling of system messages. |
| 27543 | *Summary:* Flickering when a view is resized in MFC samples.<br><br>The proper window class is defined for OpenGL window within overridden method `Cview::PreCreateWindow()`. |
| 27827 | *Summary:* Fix compilation errors in XAML (UWP) sample.<br><br>Check of export to VRML format has been added to sample XAML (UWP). `Freetype.dll` has been added to the sample project for correct execution of sample from Visual Studio. |
| 27880 | *Summary:* Fix handling of Unicode paths within MFC import/export sample.<br><br>MFC samples properly handle Unicode file names. |
| 28064 | *Summary:* Raytracing does not work in Qt samples.<br><br>`Qaction` signal `activated()` has been changed to signal `triggered()` to support Qt5. |
| 28066 | *Summary:* MFC OCAF sample could not read/save any document without environment variables.<br><br>Persistence libs are loaded using `DefineFormat()` in OCAF sample. |
| 28068 | *Summary:* Add dimension functionality shows incorrect tip for an Edge in MFC Viewer3d sample.<br><br>The warning message has been updated to show actual information.<br>Additional checking of shape type has been added to prevent exceptions for Add Dimension functionality. |
| 28105 | *Summary:* HLR rendering crash in MFC sample.<br><br>`User_Cylinder` class has been updated to work as a single object in `Prs3d_Presentation`. |

## Configuration

| | |
|---|---|
| 26846 | *Summary:* `genproj.tcl` – create hard-links instead for header redirection.<br><br>The build guides now advise using symbolic or hard links in `inc` instead of dummy header files redirecting to original one. In this case code the navigation is not messed up with extra files. |
| 27195 | *Summary:* Cmake – folders `data` and `samples/tcl` are installed with DRAW executable only.<br><br>The installation of folders `data` and `samples/tcl` has been bound with `DRAWEXE`. |
| 27212 | *Summary:* Make Cmake configurator flexible concerning option `BUILD_WITH_DEBUG`.<br><br>The option `BUILD_WITH_DEBUG` has become flexible, i.e. it allows generating Release mode without `OCCT_DEBUG` and Debug mode with it. |
| 27258<br>28027 | *Summary:* Generate built-in replacement for mandatory resource files.<br><br>The mandatory resources have been embedded into OCCT source code via a procedure for generating updated source code files from original text files.<br><br>See the details in New Features section. |
| 27287<br>27321 | *Summary:* Check for Doxygen during the first configuration and turn on overview building if Doxygen is found.<br><br>The building procedure now implicitly searches for Doxygen and turns Overview generation on if the search is successful. |
| 27339 | *Summary:* Cmake install should launch generation of overview documentation<br><br>The documentation is now generated automatically during build/install process. Overview project regenerates documentation if any source file has been changed. |
| 27344 | *Summary:* Cmake – bugs with detecting third-party products.<br><br>Information messages for TCL/TK search have been corrected.<br>Compile definitions have been added to `OpenCASCADEConfig.cmake` file. |
| 27350<br>27829<br>27861<br>27963 | *Summary:* Support Universal Windows Platform.<br><br>OCCT Cmake build procedure has been adapted for Universal Windows Platform (UWP). UWP has become an additional platform for release preparation.<br><br>See the details in New Features section. |
| 27351 | *Summary:* Custom OCCT DLL name suffixes.<br><br>It is now possible to append the postfix to names of output libraries. The postfix is defined using `BUILD_SHARED_LIBRARY_NAME_POSTFIX_DESCR` variable. |
| 27365 | *Summary:* Cmake – file writing error appears if 3RDPARTY_DIR is empty.<br><br>`3RDPARTY_DIR` and `INSTALL_DIR` are quoted in `TO_CMAKE_PATH` command to avoid error when a variable is empty. |

| | |
|---|---|
| 27369 | *Summary:* Assistance building Parasolid importer.<br><br>`OpenCASCADE${OCCT_MODULE}Targets` are now generated for all dependent toolkits. |
| 27379 | *Summary:* Unable to build `TKXDESTEP` using WOK (`wgenproj` command) in the current master.<br><br>The toolkit `TKXDESTEP` now depends on `TKShHealing`. |
| 27380<br>27461<br>27436 | *Summary:* Cmake should not try to link against 3rd-party libs when configuring a static OCCT build.<br><br>Configuration procedure for a static OCCT build now avoids linking against 3rd-party libraries and math library. Cmake GUI variables unnecessary for static build have been removed. |
| 27401 | *Summary:* Plugin – allow using statically linked plugins.<br><br>Cmake script now defines `OCCT_NO_PLUGINS` for static builds, so that linking with OCAF plugins does not lead to duplicated symbols. |
| 27402 | *Summary:* Cmake – fix compilation with `MinGw` and additional libraries.<br><br>`CSF_GL2PS` is set in `occt_csf.cmake` in the same way as `CSF_FREETYPE`.<br>Redundant `CSF_GL2PS` definition for non-Win32 configurations has been removed. |
| 27417 | *Summary:* Choose a Better Default Release Optimization Parameter for `MinGW-w64`.<br><br>The default release optimization option has been set to O2 instead of O3, since in some OCCT related examples this gives significantly smaller binaries at comparable performance with `MinGW-w64`. |
| 27424 | *Summary:* `genconf.tcl` – add missing option to enable `TKD3Dhost` in configurator.<br><br>New GUI configuration option `HAVE_D3D` allows using Direct3D on Windows. |
| 27425 | *Summary:* Unable to build AIS package with `OCCT_DEBUG` enabled in the current master.<br><br>Obsolete field `mySelName` has been removed from `AIS_LocalContext`. |
| 27458 | *Summary:* `Genproj.tcl` – add `-rpath-link` option to linker in `Code::Blocks` project.<br><br>The option `-rpath-link` has been added to linker in `Code::Blocks` project for checking the availability of all symbols in the library. |
| 27478 | *Summary:* Avoid inclusion of `Xlib.h` within `Graphic3d_GraphicDriver.hxx`.<br><br>The inclusion of `Aspect_DisplayConnection.hxx` in `Graphic3d_GraphicDriver.hxx` has been replaced by its forward declaration. |
| 27499 | *Summary:* Cmake – add option `INSTALL_NAME_DIR` on OS X.<br><br>It is now possible to specify the root of a relative installation path though `INSTALL_NAME_DIR` option on OS X. |

| | |
|---|---|
| 27513 | *Summary:* Cmake – Option `INSTALL_TBB` does not install `tbbmalloc.dll`.<br><br>Installation of `tbbmalloc.dll` has been corrected. |
| 27514 | *Summary:* Cmake – relative paths are not correctly handled for `INSTALL_DIR`.<br><br>A relative path now can be used as `INSTALL_DIR`. |
| 27566 | *Summary:* Define `Handle_` as non-template class for compatibility with C++/CLI.<br><br>`Handle_Class` types are defined for MS Visual C++ 12 and above as true classes (inheriting the corresponding `opencascade::handle<Class>`) to allow using them in `public` statement of C++/CLI language and to make these classes recognizable by other C++/CLI libraries. |
| 27639 | *Summary:* Add the possibility to build OCCT 7X without TCL/TK.<br><br>The files draw.bat/sh have stopped being generated and installed to directories `/binary` and `/install` if `DRAWEXE` toolkit is excluded from build. |
| 27661 | *Summary:* Add `genproj` for Linux in Products.<br><br>`Genproj` script has been provided for Linux in Products repository. |
| 27686 | *Summary:* Products are not compiled on ARH Linux 463 64 bit.<br><br>The problem with compilation on ARH Linux 463 64 bit has been eliminated. |
| 27749 | *Summary:* Drop remaining occurrences of GLU dependency.<br><br>Remaining occurrences of GLU dependency have been removed from templates. |
| 27754<br>27801 | *Summary:* Fix compilation of `Standard.cxx` on non-x86 desktop systems.<br><br>The problems with compilation of `Standard.cxx` on non-x86 desktop systems have been eliminated. |
| 27960 | *Summary:* Fix compilation of `OSD_Directory` with MinGW-w64<br><br>`_NATIVE_WCHAR_T_DEFINED` is checked only within `_MSC_VER` since it is msvc-specific. |
| 28077 | *Summary:* Add option for `genproj` to generate UWP projects.<br><br>The command `genproj` now can generate UWP solution when VC version is set to `vc14-uwp`.<br><br>Visual Studio project option "Use precompiled headers" is set to "Not Using" by default. Character set option of VS project Unicode is used instead of pre-processor definitions for UWP projects. |

## Coding

| | |
|---|---|
| 00176<br>25448 | *Summary:* Remove useless header `Quantity_Color_1.hxx`.<br><br>The content of useless header file `Quantity_Color_1.hxx` has been moved to `Quantity_Color.cxx`. The meaning of special `Hue` value -1.0 has been added to `Quantity_Color` methods description. |
| 24073 | *Summary:* Ambiguous call to overloaded function `StdPrs_DeflectionCurve::Match()`.<br><br>Default parameter values have been removed from the method `StdPrs_DeflectionCurve::Match()` to avoid ambiguity. |
| 24553 | *Summary:* Deleting obsolete/unused GXX files from `GCPnts`.<br><br>The file extension GXX has been changed to PXX to avoid copying files to `/inc` folder.<br><br>The code of inline functions contained in LXX files has been merged to HXX files, LXX files have been removed. |
| 27275 | *Summary:* Unused formal parameter in `BsplCLib::EvalBSplineBasis`.<br><br>Unused formal parameter has been removed from some classes in `BsplCLib` and `FairCurve` packages. |
| 27385 | *Summary:* Assertion in a static initializer in Windows GUI application.<br><br>Useless global pointer to `cerr` has been removed from `OSD_Error` together with the methods supporting its customization (Windows-only).<br><br>Global accessors to environment variables have been replaced by locals in `UnitsAPI.cxx`. |
| 27435 | *Summary:* Compilation error on Linux platform in Debug mode with enabled `OCCT_DEBUG` definition.<br><br>A handle to `Geom2d_Curve` has been added in method `MAT2d_Tool2d::Dump`. |
| 27525 | *Summary:* Eliminate warnings on Windows for OCCT with static type of libraries<br><br>Useless `*.cxx` files have been removed to eliminate linker warning `LNK4221`. |
| 27545 | *Summary:* Remove unused field in `V3d_View::myCamera`.<br><br>Unused field has been removed from `V3d_View` class. |
| 27562 | *Summary:* Avoid exporting of inline methods.<br><br>The code has been reviewed to remove `Standard_EXPORT` when it is used for inline methods. |
| 27643 | *Summary:* Eliminate GCC warning `-Wmaybe-uninitialized`.<br><br>GCC warning `-Wmaybe-uninitialized` has been eliminated in classes `GeomFill_SectionPlacement` and `GCPnts_AbscissaPoint`. |

| | |
|---|---|
| 27684 | *Summary:* Drop unused declarations from `Graphic3d`.<br><br>Some headers have been removed from `Graphic3d` package. Unused default aspects have been removed from `Graphic3d_StructureManager`. |
| 27718 | *Summary:* Class `GeomPlate_CurveConstraint` contains two same constructors.<br><br>Two constructors of class `GeomPlate_CurveConstraint` have been combined to one. |
| 27900 | *Summary:* Drop redundant Name parameter from V3d_Viewer constructor.<br><br>The following modifications have been introduced to remove `theName` parameter:<br>▪ Unused methods `AIS_InteractiveContext::DomainOfMainViewer()`, `V3d_Viewer::NextName()` and `V3d_Viewer::Domain()` have been removed.<br>▪ New methods `Contains()` and `Remove()` from `NCollection_List` take item value.<br>▪ `V3d_Viewer` now uses `V3d_ListOfLight` and `V3d_ListOfView` instead of `V3d_ListOfTransient`.<br>▪ Inaccessible methods in `NCollection_List<Message_Msg>` template are ignored. |
| 27931 | *Summary:* Coding rules, `Precision.hxx` – eliminate dead code within `PRECISION_OBSOLETE` macros.<br><br>The obsolete code has been replaced in `Precision.hxx`. `Precision.lxx` has been removed. |

Open CASCADE Technology & Products

## Documentation

| | |
|---|---|
| 25144 | *Summary:* Consider rules for formatting pointers and constants placement in expressions.<br><br>The preferred style for writing pointers declaration has been specified in coding rules developer's guide. |
| 25560 | *Summary:* Describe changes in `AIS_InteractiveConnect` and `AIS_MultipleConnected`.<br><br>New sections about scene-graph hierarchy and instancing have been added in the Visualization guide. |
| 27229 | *Summary:* `Extrema` documentation is out of date.<br><br>The documentation about Extrema functionality has been updated. |
| 27542 | *Summary:* Minor corrections in upgrade guide.<br><br>Upgrade guide has been updated. |
| 27610 | *Summary:* Spelling mistakes in the documentation about building with Cmake.<br><br>Spelling mistakes have been corrected in the documentation about building with Cmake. |
| 27631 | *Summary:* license.md duplicated in the current master HEAD.<br><br>Redundant license file has been removed from OCCT sources. |
| 27828 | *Summary:* Reference documentation of `AIS_Trihedron` contains invalid information about default size.<br><br>The documentation now correctly describes the default trihedron parameters. |

## Added-value components

### ACIS-SAT Import/Export

| | |
|---|---|
| 26624 | *Summary:* Incorrect generation of messages.<br><br>The messages are now shown without misleading statement about unknown message. |
| 27465 | *Summary:* Exception on translation of a .sab file.<br><br>The following improvements have been made in the translation of ACIS SAB files:<br>▪ Spline Surfaces can be trimmed with given parameters;<br>▪ PCurves are shifted according to the surfaces period alignment;<br>▪ PCurves are bound with their faces |
| 27635 | *Summary:* Incorrect translation of SAB files.<br><br>The offset sign is now taken into account during translation of SAB files. |
| 27689 | *Summary:* Incorrect result on export to ACIS.<br><br>Translation of periodic BSplines has been fixed in method `AcisData_CasCadeToAcis::Bs3CurveDef`. |
| 27690 | *Summary:* SAB files with names compression cannot be read in parallel mode/<br><br>Reading of SAB files with names compression in parallel mode has been enabled. |

### Parasolid Import

| | |
|---|---|
| 27331<br>27476 | *Summary:* Not valid result on translating the file.<br><br>Processing of short segments of a seam edge has been added in class `ShapeFix_ComposeShell`. |
| 27349<br>27460 | *Summary:* XtControl_Reader is not thread-safe.<br><br>The product code has been revised to get rid of thread-unsafe static variables. |
| 27723 | *Summary:* Sphere from Union lost when converting from Parasolid to Open CASCADE.<br><br>The translation of spherical faces with holes has been updated in class `XtToTopoDS_TranslateFace`. |
| 27730 | *Summary:* Parasolid "solid" generates "wire" BREP after conversion.<br><br>Method `XtToTopoDS_TranslateFace::Build` now adds some PCurves on the translation using information from file. |
| 27824 | *Summary:* Parasolid XMT file generates weird geometry after conversion.<br><br>The adaptor has been updated for C0 offsets. |

| | |
|---|---|
| 27887 | *Summary:* XMT file causes stack overflow.<br><br>Memory release has been fixed for the map of entities from `XtData_Model`. |
| 27938 | *Summary:* Sphere lost from union in XMT file.<br><br>The algorithm for translation of Spherical faces has been updated in method `XtToTopoDS_TranslateFace::Build`. |
| 27971 | *Summary:* Intersection with degenerated conical surface is translated incorrectly<br><br>Check for infinite surfaces has been added in method ShapeFix_ComposeShell::Init. |

## DXF Import / Export

| | |
|---|---|
| 27539 | *Summary:* Improve DXF Reader to read and write mesh data.<br><br>DXF Import \ Export interface has been extended with new classes `DxfEnt_Mesh` and `DxfFile_RWMesh` to read and write entities that hold mesh data. |

## Canonical Recognition

| | |
|---|---|
| 27702 | *Summary:* Highlighting of canonical geometry does not work in the Canonical Recognition sample.<br><br>`AIS_ColoredShape` has been reused in the Product to color sub-shapes. |

## Surfaces from Scattered Points

| | |
|---|---|
| 27694 | *Summary:* Installer of the sample application cannot be created.<br><br>The distribution has been corrected for the SSP Installer in Debug mode and updated for the CMake building procedure. |

## Mesh Framework

| | |
|---|---|
| 27636 | *Summary:* OMF sample is not compiled on VC++2013.<br><br>The case of obsolete handle notation has been fixed in the header file `OMFXCAF_Application.h` |
| 27653 | *Summary:* Incorrect result of STL file reading.<br><br>The ASCII STL reader has been improved to read files with empty strings between facets and skip facets composed of less than 3 vertices. |
| 27918 | *Summary:* Extend the OMF `OBJ` reader for polygons of more than 4 vertices.<br><br>The `OBJ` reader has been extended to read polygons containing more than 4 vertices. |
| 28100 | *Summary:* Extend OMF BOs to BO split and group.<br><br>New "Split and group" Boolean operation `OMFBool_SPLITandGROUP` splits the first argument shape and groups its parts depending on their location relatively to the second argument shape.<br><br>It has become possible to optionally obtain only triangular elements as a result of OMF Boolean operation.<br><br>The OMF mesh type has been extended by the algorithms to:<br>▪ copy all surface elements from one mesh to another;<br>▪ replace a mesh by its submeshes in the mesh parent.<br><br>The OMF algorithm to write triangulation from the given shape to a mesh has been extended by the option to group the triangles of each shape face to the same submesh of the mesh.<br><br>Correspondingly, the Draw command `MFshapetomesh` has been extended to optionally:<br>▪ append the result mesh by the shape triangulation,<br>▪ group the triangles to submeshes by the shape faces. |

## Express Mesh

| | |
|---|---|
| 27415 | *Summary:* Express Mesh passes a face with forward orientation to mesh adaptor, which leads to flipping of normals in case of `OMFQM_IMeshFace`.<br><br>The method `QMShape_Tessellator::TessellateFace` now passes the correctly oriented face to `MeshFaceAdaptor`. |
| 27589 | *Summary:* Improve meshing performance.<br><br>The following improvements have been introduced to improve the performance:<br><br>▪ Local data structures of method `QMBgr_FacetBuilder::VerifyAndSplitFacets` and the constructor of `QMTools_Classifier2d` have been optimized.<br>▪ Initialization of unused classifier instance has been removed from method `QMBgr_QuadTree::PostProcess`.<br>▪ Results of `BRepCheck_Edge` are cached to suppress repeated computation on the same edge in the method `QMShape_Tessellator::ComputeDeflection`. |
| 27881 | *Summary:* Express Mesh access violation in parallel mode.<br><br>Iteration by sequence iterator has been implemented in method `QMShape_Tessellator::TessellateFace` to avoid data races. |
| 27951 | *Summary:* `EMesh` creates holes.<br><br>The tolerances are used in `QMBgr_QuadNode` and `QMBgr_QuadTree` to get positions of each two tree quads relative to each other. |
| 28021 | *Summary:* EMesh divides a polygon by a link located out of the shape.<br><br>`QMBgr_FacetBuilder::VerifyAndSplitFacets()` algorithm avoids removing loops to prevent appearance of free links.<br><br>`QMBgr_QuadTree::IsEqual` checks 2d angles between a boundary node and neighboring segment for precise classification. |

## Advanced Samples & Tools

| | |
|---|---|
| 26285 | *Summary:* Update of Data Framework Browser.<br><br>A sample for `DFBrowser` can open Binary (*.cbf) and Xml (*.xml) documents. |
| 26864 | *Summary:* Introduce Eclipse IDE project for Java sample.<br><br>An Eclipse IDE project has been created for Java Import/Export sample. |

| | |
|---|---|
| 27582 | *Summary:* C# wrapper - wrap `Graphic3d_Vec4` properly.<br><br>`TKV3d.i` has been modified to wrap template-based classes completely. |
| 27657 | *Summary:* C# wrapper - fix crash when using `V3d_View::ToPixMap()`.<br><br>Manual wrapping has been implemented because `V3d_View::ToPixMap()` takes `Image_PixMap` as a non-handle reference and thus automatic wrapping produces broken code |
| 27752 | *Summary:* Advanced OCAF Sample (MFC-based) raises exception during attempt to save a document.<br><br>The export settings have been updated in OCAF Sample. |
| 27812 | *Summary:* All sub-shapes are displayed after opening document in and there is no possibility to display/erase.<br><br>The problem with display of objects has been fixed. |
| 27864 | *Summary:* Add VTK support to products Java wrapper.<br><br>The redefinition `IVtkVTK_EXPORT` error that arises on wrapping in products Java sample has been fixed. |
| 27914 | *Summary:* String path of Demo Sample in Java Wrapper is defined incorrectly on Linux version.<br><br>A note concerning `INSTALL_DIR_LAYOUT` has been added in `readme.txt`. |
| 28022 | *Summary:* C# wrapper - fix building wrapper with Debug version of OCCT.<br><br>It has become possible to build C# wrapper using VS2015 in debug mode. |

## Geodesic

| | |
|---|---|
| 27559 | *Summary:* Improvements in geodesic calculation algorithm.<br><br>Class `Geodesic_MeshDistances` has been improved to:<br><br>▪ use heap sort instead of shell sort for sorting windows, which improves performance;<br>▪ be able to split triangle by point inside it for treatment of converging fronts;<br>▪ use references for local variables to avoid copying data objects and handles;<br><br>Additional scripts have been added for `CAMgoffset` and `CAMgpath` testing |
| 27572 | *Summary:* Crash in Pocketing algorithm.<br><br>Method `PolyAlgo_MakePocket::insertAddLines` shifts the iterator to the correct position before it is used for the first time. |

# Supported Platforms and Pre-requisites

Open CASCADE Technology is supported on Windows (IA-32 and x86-64), Linux (x86-64), Mac OS X (x86-64), Android ARMv7 and x86, and iOS ARMv7 platforms.
The table below lists the product versions used by OCCT and its system requirements.

The most up-to-date information on Supported Platforms and Pre-requisites is available at http://www.opencascade.com/content/system-requirements.

| | |
|---|---|
| **Linux Operating System** | Mandriva 2010, CentOS 5.5, CentOS 6.3, Fedora 18, Ubuntu-1304, Debian 6.0, Debian 7.0 |
| **Windows Operating System** | MS Windows 10 / 8 / 7 SP1 / Vista SP2 / XP SP3 |
| **Mac OS X Operating System** | Mac OS X 10.10 Yosemite / 10.9 Mavericks |
| **Android Operating System** | Android 4.0.3 and above |
| **iOS Operating System** | iOS 7 |
| **Minimum memory** | 512 MB, 1 GB recommended |
| **Free disk space** (complete installation) | 650 MB of disk space, or 1,4 GB if installed with reference documentation |
| **Graphic library** | OpenGL 3.3+, OpenGL ES 2.0+ |
| **C++**<br>*For Linux:*<br><br>*For Windows:*<br><br><br><br><br><br>*For Mac OS X:* | <br>GNU gcc 4.3+<br>LLVM Clang 3+<br><br>Microsoft Visual Studio 2010 SP1<br>Microsoft Visual Studio 2012 Update 4<br>Microsoft Visual Studio 2013 Update 2<br>Microsoft Visual Studio 2015<br>Intel C++ Composer XE 2013 SP1<br>GCC 4.3+ (Mingw-w64)<br><br>XCode 6 or newer |
| **TCL** (for testing tools)<br>*For Linux:*<br>*For Windows:*<br><br>*For OS X:* | <br>Tcltk 8.6.3+ http://www.tcl.tk/software/tcltk/8.6.html<br>Tcltk 8.6.3+ http://www.tcl.tk/software/tcltk/8.6.html or ActiveTcl 8.6 http://www.activestate.com/activetcl/downloads<br>Built-in Tcl/Tk 8.6+ |
| **Qt** (for demonstration tools) | Qt 4.8.6 https://download.qt.io/ |
| **FreeType** (OCCT Text rendering) | FreeType 2.4.11-2.5.5 http://sourceforge.net/projects/freetype/files/ |
| **FreeImage** (Support of common graphic formats) | FreeImage 3.17.0<br>http://sourceforge.net/projects/freeimage/files/ |
| **gl2ps** (Export of OCCT viewer contents to vector graphic file) | gl2ps-1.3.8 http://geuz.org/gl2ps/ |
| **TBB** (optional tool for multithreaded algorithms) | TBB 4.x or 5.x http://www.threadingbuildingblocks.org/ |
| **Doxygen** (optional for building documentation) | Doxygen 1.8.5+ http://www.stack.nl/~dimitri/doxygen/download.html |