Open CASCADE Technology and Products version 7.3.0 is a minor release, which includes about 350 new features, improvements and bug fixes over minor release 7.2.0.
**Highlights**

**General**
- Support of Unicode file names and software signals handling for MinGW-w64 builds
- Recommendations on fulfillment of LGPL terms in Overview documentation
- Restored compatibility with Visual Studio 2008
- Support of GBK and Big5 code pages

**Application Framework**
- Ability to redefine the stored/retrieved document version and the application name

**Modeling**
- Face Removal algorithm
- Optimized surface intersection, shape offset and Boolean operation algorithms
- Oriented BoundingBoxes (OBB)
- More complete history in the family of Boolean Operations algorithms
- Improved stability of BRepPrj_Projection algorithm

**Visualization**
- Corrected translation of single-stroke fonts into BRep
- Improved compatibility with EGL on Linux, Intel HD GPUs, Mesa OpenGL, remote desktop
- Possibility to arrange more than 8 light sources and assign them to layers
- Possibility to assign Shading Model per primitive array
- Support of custom GLSL programs with Geometry and Tessellation shaders
- Distance and size culling options for rendering large scenes
- Depth pre-pass option for rendering heavy custom GLSL programs
- Verbose frame statistics for profiling 3D Viewer performance

**Data Exchange**
- Documentation for PMI in XCAF
# Table of Contents

**New Features**

- Face Removal Algorithm 4
- Oriented Bounding Boxes 4
- Generated Elements after Booleans 4

**Modifications**

- Foundation Classes 5
- Application Framework 7
- Modeling Data 9
- Modeling Algorithms 10
- Shape Healing 20
- Visualization 21
- Data Exchange 28
- Draw 30
- Mesh 31
- Samples 31
- Documentation 33
- Configuration 34
- Coding 36

**Supported Platforms and Pre-requisites** 39
New Features

Face Removal Algorithm
OCCT 7.3.0 introduces new Face Removal algorithm for 3D models. The algorithm removes features, such as holes, protrusions, gaps, chamfers, fillets, etc. from the model and fills them by extension of the faces adjacent to the feature. I.e., the feature is pulled up or down until it is completely gone (enclosed by the neighboring faces).

On the API level the algorithm is implemented in the class BRepAlgo_Defeaturing. The actual removal of faces is performed by the low-level algorithm BOPAlgo_RemoveFeatures.

Oriented Bounding Boxes
The new class Bnd_OBB stores the definition of an Oriented Bounding Box of a geometric entity. Since an Oriented Bounding Box is usually tighter than Axes-Aligned Bounding Box its usage allows producing more efficient algorithms of collision detection.

Two approaches are used to compute OBB of a shape. One is based on di-tetrahedron algorithm and is used for shapes containing triangulation or fully consisting of planar surfaces. Another approach is based on Principal Component Analysis algorithm and is used for all other shapes.

In Boolean operations the oriented bounded boxes are available in experimental mode and can be turned on by a special option SetUseOBB(flag).

Generated Elements after Booleans
Since OCCT 7.3.0 it is possible to obtain the elements Generated during Boolean operation - the new shapes obtained as a result of pure intersection (not overlapping) of the argument shapes. In the context of Boolean Operations the Generated shapes are always:
- Vertices created from the intersection points and may be Generated from edges and faces only;
- Edges created from the intersection curves and may be Generated from faces only.

So, only EDGES and FACES can have information about Generated shapes. For all other types of argument shapes the list of Generated shapes will be empty.

This feature has been implemented as the method BOPAlgo_Builder::Generated.
## Modifications

### Foundation Classes

<table>
<thead>
<tr>
<th>Modification Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>28931</strong></td>
<td>Summary: Eliminate dependency from TBB in <code>OSD_Parallel</code> header.&lt;br&gt;Methods <code>OSD_Parallel::For()</code> and <code>ForEach()</code> have been implemented in CXX files to avoid direct dependency of client code that uses <code>OSD_Parallel</code> on TBB headers and the necessity to link with TBB explicitly.&lt;br&gt;Runtime polymorphism (virtual methods) is used to hide implementation (TBB or threads-based).</td>
</tr>
<tr>
<td><strong>29064</strong></td>
<td>Summary: Copying of empty <code>NCollection</code> map takes excessive memory.&lt;br&gt;Resizing of <code>NCollection</code> maps is skipped in <code>Assign()</code> methods if the given map is empty.</td>
</tr>
<tr>
<td><strong>29171</strong></td>
<td>Summary: C signal handler does not work on MinGW.&lt;br&gt;Setting signal handler has been enabled in <code>OSD::SetSignal()</code> for MinGW (works only for SEH builds of MinGW, not for SJLJ builds).&lt;br&gt;Due to the absence of function <code>_set_se_translator()</code> in MinGW, the handler is set using C signal() function and thus is called asynchronously.&lt;br&gt;Macro <code>OCC_CONVERT_SIGNALS</code> has been enabled for MinGW build to support conversion of signals to C++ exceptions using long jumps (in the same way as on Linux).</td>
</tr>
<tr>
<td><strong>29258</strong></td>
<td>Summary: Provide move constructors for string classes.&lt;br&gt;New macro <code>OCCT_NO_RVALUE_REFERENCE</code> disables methods using move semantics on obsolete compilers that do not support rvalue references.&lt;br&gt;Method <code>Swap()</code>, move constructor, and move assignment operator have been added in classes <code>TCollection_AsciiString</code>, <code>TCollection_ExtendedString</code> and <code>NCollection_UTFString</code>.</td>
</tr>
<tr>
<td><strong>29289</strong></td>
<td>Summary: Wrong derivatives in <code>math_TrigonometricFunctionRoots.cxx</code> file.&lt;br&gt;New class <code>math_TrigonometricEquationFunction</code> has been implemented instead of <code>MyTrigoFunction</code> to provide possibilities for unit testing.&lt;br&gt;Expressions for derivatives have been corrected.&lt;br&gt;New Draw command <code>intconcon</code> provides intersection of 2d conic curves.</td>
</tr>
<tr>
<td><strong>29299 29315</strong></td>
<td>Summary: <code>NCollection</code> - define explicit empty constructor for map classes.&lt;br&gt;Ambiguous constructors have been marked with explicit keyword for classes <code>NCollection_DataMap</code>, <code>NCollection_DoubleMap</code>, <code>NCollection_IndexedDataMap</code>, <code>NCollection_IndexedMap</code>, <code>NCollection_List</code>, <code>NCollection_LocalArray</code>, <code>NCollection_Map</code>, <code>NCollection_Sequence</code>, <code>NCollection_SparseArray</code> and <code>NCollection_UBTree</code>.</td>
</tr>
<tr>
<td>Page</td>
<td>Summary</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 29302 | **Summary:** NCollection - optimize iteration of indexed maps. NCollection_IndexedMap and NCollection_IndexedDataMap now access Key by Index number without computing Hash code. IndexedMapNode::myNext2 and IndexedDataMapNode::myNext2 fields have been removed, so that indexed map now may use less memory. TCollection::NextPrimeForMap() has been extended up to 203843745 (almost full signed 32-bit integer range), and NCollection_BaseMap::mySaturated property has been removed. Duplicating checks for out of range input have been removed from NCollection_IndexedDataMap::RemoveFromIndex(), FindKey(), FindFromIndex() and ChangeFromIndex() -.

| 29344 | **Summary:** TCollection_AsciiString - replace confusing strncpy with memcpy. The use of strncpy within TCollection_AsciiString has been replaced by memcpy, where string length has been already determined. TCollection_AsciiString(const char*, int) and TCollection_AsciiString::SetValue() have been modified to throw exception on the attempt to define invalid length of the string.

| 29349 | **Summary:** OSD_Timer - add missing theThisThreadOnly constructor option available in OSD_Chronometer. OSD_Timer constructor now has theThisThreadOnly option passed to OSD_Chronometer, which is FALSE by default. OSD_Chronometer now provides methods ::UserTimeCPU() and ::SystemTimeCPU() allowing to fetch CPU times without awkward syntax through overloaded ::Show().

| 29355 | **Summary:** OCCT 6.9.1 persistence restored in OCCT 7.2.0 not working. Auxiliary classes StObjMgt_ReadData::Object and StObjMgt_WriteData::Object have been renamed to ObjectSentry (to better reflect their nature); their constructor is made explicit to ensure that such objects are always created intentionally. These objects are instantiated explicitly in the body of relevant functions, instead of implicit creation as temporary objects when function requires such object as argument.

| 29399 | **Summary:** Optimize reading of floating point values from text strings. Function Strtod() is implemented using open source (MIT-style license) code by David M. Gay instead of strtod() provided by standard run-time library. This improves its performance by 3-10 times. Functions Atof(), Strtod(), Printf(), Sprintf() and Fprintf() are declared as extern "C" to be usable from C programs. Strtod() is used in Interface_FileReaderData::Fastof() and in RWStl_Reader to accelerate their work.

| 29447 | **Summary:** The constructor of Message_PrinterOStream mixes up cout and cerr. Use of cerr or cout is corrected in Message_PrinterOStream constructor.
Summary: OSD_Spawn contains function that is not available on iOS

Method OSD_Process::Spawn() has been removed as useless (not used across OCCT or products). Note that on Linux, macOS, and Windows standard C function system() can be used directly instead, while iOS apparently does not support spawning separate processes at all.

Summary: OSD_OpenStream - handle UNICODE file paths specifically in case of MinGW-w64.

OSD_OpenStream() now uses __gnu_cxx::stdio_filebuf extension for opening UNICODE files on MinGW when using C++ file streams. OSD_OpenStream() variant accepting filebuf returns bool (true if succeeded and false otherwise).

Checks of the stream to be opened made via calls to low-level ofstream::rdbuf() have been replaced by calls to ofstream::is_open(). The proper state of the stream is also checked.

Application Framework

Summary: FSD_File and FSD_CmpFile are almost twins.

FSD_CmpFile has become a child of FSD_File, avoiding code duplication.

Summary: Managing binary format versions is not possible for own TDF_Attributes.

CDM_Application has been extended to provide application name and version stored by BinLDi v e r D o c u m e n t S t o r a g e D r i v e r, which propagates application name and version by passing it to BinMdataSt d.

BinObjMgt_RRelocat i onTabl e now stores a handle to the header data of the file to make it accessible by binary attribute drivers.

Summary: Exception on Redo.

The order of attribute deltas has been fixed in TDF_Data to perform undo/redo operations in correct sequence.

Summary: TPrsStd_AISPresentation::AISUpdate() should not implicitly redraw 3D Viewer.

Unnecessary implicit update is eliminated.

Summary: Replace CDM_MessageDriver interface by Message_Messenger.

Messenger interface CDM_MessageDriver has been replaced by classes from package Message (e.g. Message_Printer replaces CDM_MessageDriver, Message_PrinterOStream replaces CDM_COutMessageDriver).

Summary: Optimization of TPrsStd_AISPresentation::SetSelectionMode().

The method TPrsStd_AISPresentation::SetSelectionMode() has been optimized to avoid unconditional redisplay of the interactive object on change of selection mode.
<table>
<thead>
<tr>
<th>Summary</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary: The problem with the attributes constructor call.</td>
<td>The default GUID is now assigned to the attribute instead of the null GUID if the attribute constructor and AddAttribute method are used instead of Set method call.</td>
</tr>
<tr>
<td>Summary: TDat aSt d _ n t PackedMap lacks interface to set map as TCol St d _ PackedMapOf n t eger.</td>
<td>New method ChangeMap (const TCol St d _ PackedMapOf n t eger &amp; theMap) sets map available as TCol St d _ PackedMapOf n t eger.</td>
</tr>
<tr>
<td>Summary: In OCCT7.2.1-dev the names written into FSD_Fi le are associated with wrong shapes.</td>
<td>The problem with incorrect Roots indexing has been fixed by changing Dat aMap to I ndexedDat aMap. Draw command f s dr ead now allows restoring of shapes with preserved names.</td>
</tr>
<tr>
<td>Summary: Old persistence - wrong implementation of writing a reference.</td>
<td>St dObj Mgt _ W i t eDat a supports the old persistent format.</td>
</tr>
<tr>
<td>Summary: It is not possible to store Ext Str i ngArray OCAF attribute to any previous version in XML file format.</td>
<td>XmlMDataStd_ExtStringArrayDriver now allows saving to earlier XML versions.</td>
</tr>
<tr>
<td>Summary: Failed to read an OCAF XML document with 1. #QNAN value.</td>
<td>Method Xnt Obj Mgt::Get Real () has been improved to recognize NAN and infinity written by old MSVC runtime (like 1. #QNAN and 1. # NF) and detect situation when there are some trailing non-space symbols after the real value, returning False in such case. Reading of real-value attributes (single real, array, list) from OCAF XML format has been improved to create a valid attribute even if the parsing of some members fails. A warning is generated instead of error in such case.</td>
</tr>
<tr>
<td>Summary: Protection of attributes retrieval against zero ID in OCAF XML.</td>
<td>Attributes supporting several supporting user-defined IDs have been protected against zero ID in several classes of Xnt MDa t aSt d package.</td>
</tr>
<tr>
<td>Summary: Impossible to attach existing tessellation to XCAFDoc_Note.</td>
<td>New transfer object XCAFNot eObj ect s _Not eObj ect for auxiliary data contains text and attachment positions, note plane and tessellated presentation. Get Obj ect / Set Obj ect methods have been added to XCAFDoc_Not e attribute. The orientation of notes can be imported from XCAF.</td>
</tr>
<tr>
<td>Issue</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>29669</td>
<td>Crash on opening a document with the same OCAF attributes with different IDs. Protection against clash of attributes with the same GUID while opening the document from a binary or XML file has been provided in the case when more than one attribute of the same type but with different GUIDs are stored on the same label.</td>
</tr>
<tr>
<td>29816</td>
<td>Add the possibility to get/set shape presentations for GD&amp;T label through one function. New methods <code>GetGDTPresentations()</code> and <code>GetGDTPresentations()</code> have been added for Dimension Tool XCAFDoc_DimTol Tool.</td>
</tr>
</tbody>
</table>

**Modeling Data**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>27356</td>
<td>BRepTools::Clean() does not clean free edges from Poly_Polygon3D. BRepTools::Clean() now cleans 3D polygons on edges.</td>
</tr>
<tr>
<td>29102</td>
<td>Missing points using GCPnts_QuasiUniformDeflection. The method <code>GCPnts_QuasiUniformDeflection::PerformCurve</code> provides proper derivative at the end of the current curve. Draw commands <code>crvpoints</code> and <code>crvtpoints</code> now work with wires as composite curves.</td>
</tr>
<tr>
<td>29287</td>
<td>Move package GProp from TKG2d to TKG3d.</td>
</tr>
<tr>
<td>29448</td>
<td>The method <code>Extrema_FuncExtCS::GetStateNumber</code> mixes up parameter on curve with parameter U on surface. The code has been corrected to avoid confusion.</td>
</tr>
<tr>
<td>29775</td>
<td>BRepAdaptor or _CompCurve parameterization is incorrect. It has been explicitly stated in BRepAdaptor or _CompCurve that this class can only work on valid wires where all edges are connected to each other to make a chain.</td>
</tr>
</tbody>
</table>
### Modeling Algorithms

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong> Unify faces classification procedures in Boolean Operations.</td>
<td>New method <code>BOPAlgo_Tools::ClassifyFaces()</code> provides a unified face classification procedure for methods <code>BOPAlgo_Builder::FillIn3DParts()</code> and <code>BOPAlgo_BuilderSolid::PerformInternalShapes()</code>. BOP intersection algorithm <code>BOPAlgo_PaveFiller</code> now forces intersection of edges after the tolerance values of their vertices have been enlarged. <code>BOPAlgo_Tools::PerformCommonBlocks()</code> method now avoids losing faces of already created Common blocks. When PCurves are built for edges on faces, the validity of existing PCurves is checked for periodic surfaces and PCurves are adjusted if necessary.</td>
</tr>
<tr>
<td><strong>Summary:</strong> Prism from BSpline curve cannot be chamfered.</td>
<td>Methods <code>IntCurveSurface_ComputeTransitions</code> and <code>IntTools_EdgeEdge::IsIntersection</code> have been protected against zero-length vectors.</td>
</tr>
<tr>
<td><strong>Summary:</strong> Result of blend fails the bopcheck.</td>
<td>Protection against zero-length <code>gp_Dir</code> construction has been added in method <code>El_Cli b::Li neParame ter</code>.</td>
</tr>
<tr>
<td><strong>Summary:</strong> Thickness fails on cylinder with draft.</td>
<td>Calculation of intersection in 2D space has been corrected in class <code>BRepOffset::set_Tool</code>.</td>
</tr>
<tr>
<td><strong>Summary:</strong> Clean up the duplicate classes in TKBO project.</td>
<td>Collection classes from <code>BOPCol</code> package have been replaced by classes from <code>TopTools</code> and <code>TColStd</code> packages or removed. Additionally:  - The class <code>BOPDS_PassKey</code> and its containers have been removed as unused;  - The containers <code>IntTool_s_0 target aMapOf ShapeAddres s</code> and <code>IntTool_s_1 ndexedDat aMapOfTransient Address</code> have been removed as unused;  - The container <code>BOP_0 target e_Dat aMapOf ShapeBox</code> is replaced with <code>TopTool_s_0 target aMapOf ShapeBox</code>;  - The class <code>BOPTool_s</code> has been removed as a duplicate of the class <code>TopExp</code>.</td>
</tr>
<tr>
<td><strong>Summary:</strong> Replacement of old Boolean operations with new ones in <code>BRepProj_Projection</code> algorithm.</td>
<td>The use of <code>BRepAlgo_Section</code> has been replaced with <code>BRepAlgoAPI_Section</code> in <code>BRepProj_Projection</code> algorithm. The performance of <code>BRepAlgoAPI_Section</code> has been improved. The API classes from package <code>BRepAlgo</code> that provide access to old Boolean operations are marked as deprecated. The corresponding classes from package <code>BRepAlgoAPI</code> should be used instead.</td>
</tr>
</tbody>
</table>
| 26570 | **Summary:** Crash on attempt to rotate a shape.  
Draw-command `trotate (ttranslate, tmirror...)` has been extended by additional parameter `-copy`.  
New check of edge range has been added in `BRepCheck`, `BRepCheck_Edge` and `ShapeAnalysis_Edge`.  
`BRepTools_TrsfModification::NewCurve2d()` and `GeomLib::SameRange()` have been modified to avoid exception in `TrimmedCurve`. |
| 26677 | **Summary:** Boolean fuse operation produces incorrect result.  
The procedure of initialization of `BRepTopAdaptor_FClass2d` and `IntTools_FClass2d` classifiers has been corrected to produce a tighter polygon in case of self-intersections on very thin faces. |
| 26789 | **Summary:** invalid result of BOP Fuse.  
Solid Builder algorithm (`BOPAlgo_BuilderSolid`) now avoids creation of INTERNAL solids from unclassified faces.  
Instead the user is warned that some of the faces have been unclassified and not used for solids creation. |
| 28102 | **Summary:** Problem cutting a plate with several holes (670).  
`IntPatch_ImpPrmIntersection` algorithm now correctly processes cases when the point of Walking-line splitting is near to the boundary of the intersection domain but does not match this boundary. |
| 28150 | **Summary:** Exception is raised during Boolean operation  
Method `ProjLib_ComputeApproxOnPolarSurface::BuildInitialCurve2d(....)` now uses a correct number of points. |
| 28245 | **Summary:** Result of Cells Builder algorithm becomes invalid after removal of internal boundaries on faces.  
`BOPAlgo_CellBuilder` has been corrected to remove internal boundaries between faces of the same material at once for the whole shape, while preserving the boundaries between areas with different materials. |
| 28248 | **Summary:** HLR algorithm result is retrieved from the last added shape only.  
Mistakes in retrieving of `HLRBRep_EdgeData` by value instead of reference have been corrected. |
| 28385 | **Summary:** Improve drawing isolines (`DBRep_IsoBuilder` algorithm).  
The algorithm `DBRep_IsoBuilder` computing the iso-lines on face for display in DRAW viewer now avoids iso-lines going out of face.  
Additional `Init()` method has been implemented for `WireExplorer` algorithm. It takes UV bounds of the face to avoid their repeated computation when a face with multiple wires is processed. |
Summary: Improve performance of Boolean Operations.

The following improvements have been introduced to increase performance of Boolean Operations on relatively fast cases:

- Facet information is initialized for the faces participating in Face/Face interference even when the gluing is ON to take into account intersection of their sub-shapes.
- Methods `BOPAlgo_ShellSplitter::MakeConnexityBlocks` and `BOPAlgo_WireSplitter::MakeConnexityBlocks` have been unified into `BOPTools_AlgoTools::MakeConnexityBlocks`.
- Unnecessary bounding box computation is avoided for solids during DS initialization. The bounding boxes for solids will be computed during the building stage to find faces located inside solids. For the shape self-interference check (performed by the `BOPAlgo_CheckerSI`), the bounding box is still computed, as it is necessary to resolve Shape/Solid intersections.
- Only three sample points are used to check coincidence of line and plane.
- Planes intersection is performed only when the gluing is off.
- Repeated initialization of 2D classifier while building splits of the faces is avoided.
- Methods `CorrectWires` and `CheckEdge` save data to avoid its recalculation.
- It is possible to disable the classification of the input solids on the inverted status (to be the holes in the space).
- Building of bounding boxes for faces/solids during splitting of the input arguments for their classification into hole faces/shells is avoided if no holes are created.
- Rebuilding of the faces/solids from arguments which does not acquire any inside parts of other arguments is avoided by using their draft versions as their splits.

Summary: Conversion of a spherical face to a spline produces an invalid shape.

2D tolerance calculation for BSpline/Bezier surfaces has been improved in `BRepCheck_Wire` class.

Summary: Intersection algorithm produces the curve with oscillation.

The algorithm `IntPatch_WlineTool` purging extra points from the walking line now avoids making too large distance between two neighbor segments of the line.

Summary: Intersection of faces gives exception in debug mode.

New method `IntWalk_Pwalking::RemoveAPoint()` provides safe removal of points from the Walking-line.

Summary: Infinite loop at intersecting two faces / surfaces.

`IntWalk_Pwalking` has been fixed to avoid constantly increasing/decreasing steps.

Summary: `BRepOffset_MakeOffset` produces invalid shape (`thick shell`) in Intersection mode.

- Method `BRepOffset_Tool::Inter3D` is now used for selection of proper edges. They are not concatenated into one edge if they pass through a vertex on boundary.
- Selection of edges has been eliminated in method `BRepOffset::Connex1 ntYl ntYl nt` because now this method is able to process seam edges correctly.
Summary: BO algorithm is stuck while fusing shell and edges.

Approximation parameters: degmin, degmax, max number of segments, boundary condition and maximal projection distance have been added in the interface of classes ProjLi b_Pro ject edCur ve, ProjLi b_ComputeApp r ox and ProjLi b_ComputeApp r oxOnPol arSu r face.

Appr ox /Appr ox_Comput eCLi ne algorithm now can treat the maximal number of segments allowed for cutting.

Method BOPTool s_AL goTool s2Dx : MakePCur veOnFace( ..) now manages cases with big edge tolerances.

Summary: 2D offset creates faulty result from wire.

IntTools_EdgeEdge intersection algorithm has been protected from incomplete type conversion caused by presence of Trimmed curves by using Adaptors providing typed curves instead of direct casting.

Summary: IntTools_EdgeEdge::FindParameters() hangs on bad curve.

The possibility to dramatically increase the step during iteration on curve when the global resolution of the curve is too small has been added in functions FindPar amet ers, FindBestS olu tion and Fi ndDi st PC.

Summary: General Cut produces invalid shape.

The reporting system of Boolean operations provides new warnings:

- BOPAlgo_AlertIntersectionOfPairOfShapesFailed when the intersection of pair of sub-shapes of the arguments has failed;
- BOPAlgo_AlertBuildingPCurveFailed when the building of the 2D curve of the edge on face has failed;
- BOPAlgo_AlertAcquiredSelfIntersection when the positioning and tolerances of the arguments lead to creation of self-interfered shapes.

These warnings allow completing the operation even if some sub-shapes do not intersect or some PCurves are not built. They also give pairs of sub-shapes, on which the intersection/projection has failed, providing the ability to analyze the intersection results.

Summary: Extra shapes in result of General Cut (box by ellipsoid).

Search for splitting parameters on degenerated edges has been improved in BOPAlgo_go_PaveFill er algorithm.

Summary: Result of Boolean common depends on the order of arguments.

Usage of Bnd_Box-filtering has been eliminated while putting a (definitely) common vertex between two faces on the intersection curve.

Summary: 3D Offset algorithm produces a NULL shape.

UpdateVal i dEdges function from BRepOf fset_MakeOffset_1 filters splits of the edges in two stages:

- Filters the connected blocks separately using localized bounding edges taken only from the splits of offset faces from the block;
- Provides combined treatment of the remaining splits using bounding edges from the splits of all offset faces.
<table>
<thead>
<tr>
<th>Number</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>29157</td>
<td>Suspicious pass-through of case labels in switch statements. Suspicious passes through case labels have been resolved by using St andar d_FALLTHROUGH macro or by redesigning the code.</td>
</tr>
<tr>
<td>29159</td>
<td>Sewing fails when using a maximum tolerance. BRepBuilderAPI_Sewing now uses the truly computed edge tolerance if the tolerance imposed by BRepLib::SameParameter is too large.</td>
</tr>
<tr>
<td>29162</td>
<td>Geom2dIntersect algorithm does not find intersection of ellipse and line. Analytical intersection algorithm has been implemented for ellipse-line intersection in lntCurve_1ntConicConic class.</td>
</tr>
</tbody>
</table>
| 29175  | BOPAlgo_PaveFiller sometimes raises exception in parallel mode. Data races are now avoided in BOPAlgo_PaveFiller:  
- Only unique edge-face pairs are processed.  
- A copy of the edge is made in each thread and updated if the same edge is treated simultaneously with different faces in different threads. The original edge is updated only when parallel processing is finished.  
New method BOPTool::CopyEdge copies edge with vertices. |
| 29185  | Move AddTool(), SetTool(s), Tool(s) and other common methods of BOP tools to a separate interface class. New BOPAlgo_ToolsProvider class with methods AddTool(), SetTool(s) and Tool(s) has been added. BOPAlgo_BOP and BOPAlgo_Splitter are now successors of BOPAlgo_ToolsProvider. |
| 29188  | Null shape is produced by 3D offset algorithm (mode="Complete", Join Type="Intersection").  
The following improvements have been made in the 3D offset algorithm for mode "Complete" and Join type "Intersection":  
- RemoveInvalidFaces() removes the invalid parts outside of the solids built from the splits of offset faces. It helps to avoid their rebuilding and to speed up the computation.  
- FindVerticesToAvoid() strengthens the criteria for the vertices to be avoided in the new splits. |
| 29204  | BRepOffsetAPI_MakePipeShell produces invalid result and raises exception in Draw. ChooseSection algorithm searching for section in the corner is now able to find simple cases with rather big tolerance.  
The constructor of BRepFill_Sect i on now removes locations in the section shape as it is done in BRepFill_Swee p::Bu i l d.  
Correction of U-edges by Same Parameter has been added to the method BRepFill::Sweep::Bu i l d.
### Summary: **BRepOffsetAPI_NormalProjection** produces internal edges and vertices

- The algorithm `BRepOffsetAPI_NormalProjection` uses section operation instead of common to get the edge-result of projection within face restrictions.
- The algorithm `ShapeUpgrade_UnifySameDomain` now correctly gathers same domain faces in a compound.
- The script `snowflake.tcl` does not depend on the order of edges in the result.

### Summary: Improve performance of Boolean Operations.

Performance of Boolean operations algorithm has been improved by:

- Improving the check of Same Domain faces in `BOPAI go_Builder::FillSameDomainFaces()`;
- Faster rejection of outer faces for solids using Bounding Box classification first in `BOPAI go_Builder::FillIn3Dparts();`;
- Using `IncAllocator` for local containers.
- Method `BOPAI go_PaveFiller::IsExistingPaveBlock()` provides correct edge tolerance.
- Method `PutClosingPaveOnCurve()` now uses the tolerance of the pave put on the bound to check curve for closeness and valid range (to avoid considering small curves (within vertex tolerance) closed).

### Summary: Using OBB to speed up Boolean Operations.

The Oriented Bounding Boxes (OBB), are the bounding volumes enclosing shapes in the same way as the Axes-Aligned Bounding Boxes (AABB). Generally, the OBB should be much tighter than AABB, thus can be used more effectively for fast rejection of non-interfering objects. The OBB objects have been implemented in OCCT and integrated into modeling algorithms as a new class `Bnd_OBB`.

By default the usage of OBB is turned off. It is enabled by the method `SetUseOBB(flag)` available for all operations in Boolean Component.

In Draw the command `buseobb 0/1` should be used. Note, that this will affect all subsequent operations.

The OBB for shapes are built by the first necessity and stored into operation context `IntTools_Context`.

### Summary: Improve performance of 3D offset algorithm for the planar cases.

Intersection of offset face splits in "Complete" mode is now performed by the dedicated method `BUI l dShell sCompl et e nt er()`.

### Summary: Prevent modification of the input shapes in case their sub-shapes have not been modified.

The following changes have been introduced to prevent modification of input shapes in destructive mode if their sub-shapes have not been modified:

- In `BOPAI go_PaveFiller::MakeSplitEdges`, edge splitting for pave blocks with old vertices is avoided if it is possible to use the existing edge.
- In `BOPAI go_Builder::FillImagesContainer` new containers (WIRES/SHELLS/COMPSOLID) are not created if no parts have been modified;
- In `BOPAI go_Builder::FillImagesFaces`, a new face is not created if no wires have been modified;
- In `BOPAI go_Builder::FillSameDomainFaces`, the original face is used as a representative for the group of SD faces, if possible.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>29351</td>
<td>Boolean Operations create invalid PCurves.</td>
</tr>
<tr>
<td></td>
<td>BOPTool s Al goTool s2D algorithm now checks if the produced 2D curve has the same range as 3D curve of the edge when it makes a PCurve for an edge on face.</td>
</tr>
<tr>
<td>29356</td>
<td>GCpnts_TangentialDeflection hangs on specific curve.</td>
</tr>
<tr>
<td></td>
<td>While UV Iso parameters are obtained in StdPrs_Isolines, UV limits should be applied only if face bounds have infinite values. Method StdPrs_Isolines::UVsoParameters has been fixed to get correct Iso lines without modification of UV limits.</td>
</tr>
<tr>
<td>29358</td>
<td>Unifysamedomain is unable to merge faces with the same underlying surface.</td>
</tr>
<tr>
<td></td>
<td>The description of Unifysamedomain algorithm has been updated to avoid misunderstanding of its behavior.</td>
</tr>
<tr>
<td>29359</td>
<td>Approximation algorithm computes multidimensional distance in Euclidean space incorrectly.</td>
</tr>
<tr>
<td></td>
<td>Wrong distance computation in case of Approx_ChordLength approximation type has been corrected.</td>
</tr>
<tr>
<td>29363</td>
<td>No history for shapes which were produced as a result of intersection.</td>
</tr>
<tr>
<td></td>
<td>The method Generated has been implemented for the algorithms in Boolean Component. A shape can be Generated only if it is a result of pure intersection (not overlapping) of argument shapes. Thus, the Generated shapes are:</td>
</tr>
<tr>
<td></td>
<td>- VERTICES created from the intersection points and generated from edges and faces only;</td>
</tr>
<tr>
<td></td>
<td>- EDGES created from the intersection edges and generated from faces only.</td>
</tr>
<tr>
<td></td>
<td>Thus only EDGES and FACES can contain Generated shapes. For all other types of shapes the list of Generated shapes will be empty.</td>
</tr>
<tr>
<td></td>
<td>Methods Modified and IsDeleted have been optimized and simplified based on the correct filling of BOPAlgo_Buil dShape: : ny! images and the BOPAlgo_BuilderShape::myImagesResult map.</td>
</tr>
<tr>
<td></td>
<td>The history of solids unification is provided by the CellsBuilder algorithm.</td>
</tr>
<tr>
<td></td>
<td>The User guide on Boolean Operations has been updated with new chapter &quot;History Information&quot; describing the rules for filling history for operations in Boolean Component.</td>
</tr>
<tr>
<td>29368</td>
<td>Incorrect intersection state of the intersection point of two 2d curves.</td>
</tr>
<tr>
<td></td>
<td>The algorithm math_FunctionRoots now uses two methods to find the function extremum (via the zero value of the derivative function and using the old approach), then it chooses the best of two solutions computed by different methods.</td>
</tr>
<tr>
<td>29387</td>
<td>Incorrect result of cutting a face with several shapes.</td>
</tr>
</tbody>
</table>
|         | Section edges without valid range are now removed by BOPAlgo_PaveFiller algorithm.
Summary: Fuse of two edges creates self-interfered shape.

Method BOPAlgo_PaveFiller::SplitPaveBlock avoids creation of edges with too small valid range (which hinders splitting the edge) and interfering vertices.

Summary: Curve evaluation at boundary point.

Previously BRepAdaptor_CompCurve considered the input wire to be periodic with period LastParameter() – FirstParameter() if it contained a single periodic edge. Now method IsPeriodic always returns FALSE because it is impossible to obtain correspondence between the members of BRepAdaptor or _CompCurve class and its periodicity status.

Summary: Method BndBox::IsOut() returns true for point lying on the planar face.

The method BRepBndLib::Add now enlarges the bounding box by the tolerance of edges whose curves participate in the calculation of the box.

Summary: Extrema_ExtCC returns IsParallel equal to true for not parallel curves.

To decide that the curves are parallel it is additionally checked in Extrema_GenExtCC if their ends do not diverge.

Summary: DRAW command splitshape produces invalid result on the cylindrical face.

The distance between edges in UV space is now taken into account for periodical surfaces in method LocOpe_SplitShape::ChoixUV.

Summary: Implementation of Face Removal algorithm.

Face Removal algorithm removes unwanted holes, protrusions, gaps, chamfers, fillets, etc. from a 3D model consisting of solids. The algorithm removes all requested features from the shape and builds the new shape as a result. The input model is not modified.

On the API level the algorithm is implemented in the class BRepAlgo_ApiDefeaturing. The actual removal of faces is performed by the low-level algorithm BOPAlgo_RemoveFeatures.

The following changes have been made in the algorithms used by Face Removal:

- History support for the solids is provided in ShapeUpgrade_UnionIfSameDomain.
- It is possible to merge History of any Algorithm with standard history methods, such as IsDeleted(), Modified() and Generated() into BRepTools_History.

Summary: Avoid inheritance of the BRepAlgo_API_Check from BRepBuilderAPI_MakeShape.

BRepAlgo_API_Check class is inherited from BOPAlgo_Options instead of BRepAlgo_API_Algo, as the latter is too excessive for checking purposes.

Draw command bopapi_check has been added for testing the BRepAlgo_API_Check algorithm.
<table>
<thead>
<tr>
<th>Summary</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean operation &quot;general fuse&quot; creates a solid containing 5 not connected shells lying on the same level.</td>
<td>BOPTools_AlgoTools::AreFacesSameDomain takes into account possible deviation of edges from the surface faces when checking two faces with the same bounds on Same Domain.</td>
</tr>
<tr>
<td>Intersection line between two parametric surfaces is restricted incorrectly if it matches the surface boundary.</td>
<td>Creation of Int Pat ch_Poi nt s is now forbidden in tangent zones except for domain boundaries of the intersected surface.</td>
</tr>
<tr>
<td>No intersection curve between faces if starting points are given.</td>
<td>Bounded Int Pat ch_Poi nt s are now found in case when starting points are used in intersection algorithm. Before the fix, these points were not looked for.</td>
</tr>
<tr>
<td>Improve performance of ShapeUpgrade_UnionSameDomain n : UnionEdges() method.</td>
<td>access violation in HLRBRep_Pol yAl go while computing HLR for triangulation-only surface.</td>
</tr>
<tr>
<td>BRepExtrema_DistShapeShape returns only one solution but the wires have two intersections.</td>
<td>The algorithm GenExt CC subdivides long curves into parts if their length is too different.</td>
</tr>
<tr>
<td>Improvements in the class BRepOffset_Tool.</td>
<td>BRepCl ass3d_Sol i dCl assi fl er classifies the point as IN while it is ON.</td>
</tr>
<tr>
<td>The algorithms of curve/face and curve/surface intersection (Int CurvesFace_Int er sect or and Int CurveSurf ace_Int er sect i on) now process simpler some analytical cases when a curve is parallel or belongs to a surface.</td>
<td>Misprint in the formula of Hessian computation in file GeomLib_CheckCurveOnSurface.cxx.</td>
</tr>
</tbody>
</table>
**Summary:** Distance between a cylinder and a straight line.

In Extrema_ExtCS::Perform the line is trimmed by corner points of surface bounding box to reduce its range.

**Summary:** Misprint in BuildEdge(…) static function of BRepFill_Sweep.cxx file.

**Summary:** Exception in BRepFill_PipeShell algorithm.

BRepFill_Sweep algorithm is improved to handle cases when generated revolution surface has degenerated point in the middle.

**Summary:** Boolean intersection with fuzzy-option hangs.

The algorithm of step re-computation has been improved for the case of two faces intersected by their boundaries.

**Summary:** Wrong result of CUT operation.

New function HasMultiConnected from BOPAlgoBuilder checks if the edge has multi-connected vertices. It can be used to check if the face split by the intersections with other arguments can be split by a vertex. In this case simple face reconstruction is avoided and BuildFace algorithm is used to split the face.

When it is checked, if the split edge is oriented as the original one in BOPTools::IsSplitToReverse(), the tangent vectors are computed for both edges at the same point. This point is taken on the split edge and projected on the original edge. It has been ensured that the reference point is taken inside the valid range of the split edge (i.e. not covered by the tolerance spheres of its bounding vertices) and can be successfully projected on the original edge. Moreover, several sampling points are now taken on the split edge and processed until the first valid point is found.

If requested, (by a non-null pointer) all BOPTools::IsSplitToReverse() methods are now return the error status of the check. Before using the returned flag, the calling program should check this error status. For a successful check the error status should be equal to zero.

New warning BOPAlgo_AlertUnableToOrientTheShape is now returned in Boolean algorithms if the check for correct shape orientation has failed.

**Summary:** BRepTools::Update(Face) unexpectedly updates UV points of PCurve.

The method BRepTools::UpdateFaceUVPoints has been modified to reset the UV points of the edge to the bounding points of the parametric curve of the edge on the face.
## Shape Healing

### 28467

**Summary:** Improve UnifySameDomain performance.

Some not needed modes of fix are now turned off in the called ShapeFix Face algorithm.

PCurves are stored on planes in the edges to avoid repeated computation of the same PCurves many times (it is done only when SafeInputMode is false).

Unnecessary replace/apply actions in the modification context are avoided.

New command buildpcurvesonplane builds and stores PCurves of edges on planar faces. This is useful for investigation how the presence of PCurves on planes influences the performance of algorithms.

The drawing of dimension line in snowflake test has become independent on the order of vertices in the result.

### 28681 29544

**Summary:** UnifySameDomain distorts face boundary when it merges a chain of small linear edges.

The function GetLineEdgePoints now takes into account linear tolerance value in case of a sequence of edges based on lines (which are unified into one line-segment).

### 29695

**Summary:** Infinite loop in ShapeFix IntersectionTool.

Splitting of a wrong edge is avoided in ShapeFix IntersectionTool.

Protection has been added for the case when the index of the edge to be split is out of range.
## Visualization

| 22048 | **Summary:** AISInteractiveContext – single object selection should always clear multiple selection.  
AISInteractiveContext::Select() now clears multiple selection when selecting a single object. |
| 26127 | **Summary:** Default camera is not copied in copy constructor of V3d_View.  
V3d_View copy constructor now copies DefaulCamera() from the specified View. New argument –cl oneActive has been added in Draw command vi ni t. |
| 27618 | **Summary:** Selection returns entity overlapped by another entity on border cases.  
- Tolerance is scaled according to Camera definition in SelectMgr_View::updatePoint3d().  
- gp::Resolution() is used instead of Precision::Confusion() in SelectMgr_RectangularFrustum::segmentSegmentDistance(). |
| 27732 | **Summary:** AIS_ConnectedInteractive crashes on NULL handle returned by MeshVS_CommonSensitiveEntity::GetConnected().  
NULL-check in AIS_ConnectedInteractive::ComputeSelection() as well as interface methods MeshVS_Mesh::AcceptDisplayMode() and MeshVS_CommonSensitiveEntity::GetConnected() have been added. |
| 28069 | **Summary:** Allow picking Graphic3d_TypeOfShadingModel per-object  
It has become possible to define shading model on per-object level. For this:  
- Graphic3d_AspectFillArea3d has been extended by new property ::ShadingModel(), which is set to Graphic3d_TOSM_DEFAULT by default. The new API allows assigning Shading Model to specific Primitive Array groups instead of the entire Viewer, which was the only possibility before.  
- Graphic3d_TypeOfShadingModel has been extended with Graphic3d_TOSM_DEFAULT value meaning that Shading Model defined as default for the Viewer should be used. Graphic3d_TOSM_NONE has been renamed to Graphic3d_TOSM_UNLIT.  
- V3d_TypeOfShadingModel enumeration has been merged into Graphic3d_TypeOfShadingModel avoiding duplicated definitions and confusion. Old values remain for compatibility with old code and can be marked deprecated in the future.  
- Draw Harness command vaspects has been extended by new argument –set ShadingModel for testing Shading Models assigned to entire objects.  
- OpenGl_SetOfShaderPrograms now holds an array of Shading Models. OpenGl_ShaderManager interface has been modified and now requires enumeration as input in several places where Boolean flags have been used previously (methods ::Bi ndFacePr ogr am(), ::Bi ndLi nePr ogr am() and ::Bi ndMa rker Pr ogr am()).  
- OpenGl_Workspace now defines default OpenGl_AspectFace as Graphic3d_TOSM_UNLIT to simplify indication of primitive groups with undefined Fill Area aspects. Graphic3d_TOSM_UNLIT set as default Shading Model avoids artifacts on Lines and Markers.  
- Missing initialization about Fill Area aspects has been added in AIS_Manipulator::Axis::Compute(). |
### Summary: AIS_ConnectedInteractive does not support exact HLR mode.

AIS_ConnectedInteractive now shares HLR computation code with AIS_Shape. BRepTool::WireOrientation() returns TopAbs_FORWARD instead of throwing exception in case of infinite Edge without vertices.

Std::select::BRepSelectionTool::GetSensitiveForFace() creates Select3D::SensitiveCurve from 2 points in case of infinite Edge instead of Select3D::SensitiveFace.

### Summary: SelectMgr_SelectionManager::Activate() should not implicitly deactivate Global Selection Mode.

Implicit deactivation of global selection mode has been removed from SelectMgr::Activate().

New method AIS::InteractiveContext::SetSelectionModeActive() replaces AIS::InteractiveContext::Activate()/::Deactivate(). This method takes an argument AIS::SelectionModesConcurrency, which defines what to do with already activated selection modes:

- AIS::SelectionModesConcurrency_Single: only one selection mode can be activated at the same moment – previously activated ones should be deactivated;
- AIS::SelectionModesConcurrency_GlobalOrLocal: either Global (AIS::InteractiveObject::GlobalSelectionMode()) or Local (multiple) selection modes can be active at the same moment;
- AIS::SelectionModesConcurrency_Multiple: any combination of selection modes can be activated.

### Summary: TKOpenGl – avoid excessive frustum culling traverse within extra OIT rendering pass.

Culling traverse is no more called implicitly within OpenGl::Layer::Render(). Instead, all layers are traversed at once within OpenGl::View::render() beforehand. 

OpenGl::BVHTreeSelector methods have been renamed to better reflect their meaning.

Non-persistent culling options have been moved to the dedicated structure OpenGl::BVHTreeSelector::CullingContext so that OpenGl::BVHTreeSelector instance can be used for different Layers without modifying its state.

### Summary: SelectMgr::Selectable::eObject – move out iterator from object.

SelectMgr::Selectable::eObject now provides access to the list of selections using external iterator objects. For this:

- New method SelectMgr::Selectable::eObject::Selections() replaces deprecated methods Init(), More(), Next() and CurrentSelection().
- New method SelectMgr::Selection::Entities() replaces deprecated methods Init(), More(), Next() and Sensitivity().
- SelectMgr::Selectable::eObject::myAssemblyOwner has been moved to AIS::MultipleConnectedInteractive.
- SelectMgr::Selectable::eObject::Selection() now returns NULL handle for not found selection.
### Summary:

- **SelectMgr_SelectableObject::HasSelection()** is no more virtual and just returns `!Selection().IsNull()`.
- **SelectMgr_SelectionManager::mySelectors** map is now declared using proper key type `Handle(SelectMgr_ViewerSelector)` instead of `Handle(Standard_Transient)`.
- Broken `HasTransformation()` checks have been removed from `SelectMgr_SelectableObject` and `SelectMgr_SelectionManager`.

### Summary:

- **TKOpenGL** – support Geometry Shader definition.
  
  *Graphic3d_TypeOfShaderObject* enumeration has been extended by Geometry shader object type.
  
  - `OpenGl_ShaderProgram::Initialize()` processes new shader object types when supported by OpenGL version.
  - Declarations.glsl has been fixed so that occFragCol or is defined only for Fragment Shader object (by handling new `FRAGMENT_SHADER` macros).
  - Draw command `vshader` has been extended to support definition of Shader Object types other than Vertex and Fragment shader.

### Summary:

- **AIS_Manipulator** – broken transformation is applied at Rotation angles near to Pi.

  Use of manipulator axes with temporarily applied transformation (when `BehaviorOnTransform::FollowRotation` is TRUE) has been fixed in `AIS_Manipulator::ObjectTransformation()`. Start axes orientation (at the beginning of Rotation) is now used instead.

### Summary:

- **SelectMgr_FrustumBuilder** constructor is not exported without arguments.

  `Standard_EXPORT` has been added in `SelectMgr_FrustumBuilder.hxx`.

### Summary:

- **AIS_Trihedron** – add option hiding arrows tips.

  `Prs3d_DatumAspect::DrawDatumPart()` now handles new flag `ToDrawArrows()`.

  New option `arrowTip` has been added in `vtrihedron`.

### Summary:

- **Improve Font_BRepFont** to handle one-line-fonts.

  New property `SingleStrokeFont()` has been added in `Font_SystemFont`. `Font_BRepFont::renderGlyph()` now does not close contours when flag `SingleStrokeFont()` has been set.

### Summary:

- **Tests** – failures when tests are executed via Remote Desktop connection.

  Protection against accessing null pointer has been added in `OpenGl_VertexBuffer::Create()`.

  OpenGL-related warnings in command `vreadpixel` are redirected to `cout` so that they do not contaminate the command output.
<table>
<thead>
<tr>
<th>29124</th>
<th>OpenGL version check before retrieving <code>GL_SHADING_LANGUAGE_VERSION</code> string has been added in OpenG_Cont ext:: Di ag nost i c:: Info r ma t i on(). Out-of-memory writing within fallback View dump mode has been fixed in V3d_Vi ew.cxx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>29137</td>
<td><strong>Summary:</strong> D3Dhost_FrameBuffer should provide software fallback when WGL_NV_DX_interop is unavailable. D3Dhost_FrameBuffer now provides fallback code copying OpenGL FBO content into D3D surface (slow). D3Dhost_FrameBuffer releases Depth texture and FBO index. FBO dump implementation has been removed from OpenG_Workspace::BufferDump().</td>
</tr>
<tr>
<td>29147</td>
<td><strong>Summary:</strong> D3Dhost_FrameBuffer::BindBuffer() fails on some Intel drivers. D3Dhost_FrameBuffer::BindBuffer() now implicitly detaches Depth+Stencil texture in case of driver failure. D3Dhost_Vi ew::d3dCreateRenderTarget() now does not request Depth+Stencil texture by default.</td>
</tr>
<tr>
<td>29158</td>
<td><strong>Summary:</strong> Suspicious pass-through of case labels in switch statements. Incorrect fallthrough from MeshVS_SMF_Mesh to MeshVS_SMF_Group has been fixed in MeshVS_Mesh::ComputeSelection().</td>
</tr>
<tr>
<td>29165</td>
<td><strong>Summary:</strong> Misuse of enumeration in Prs3d_DatumAspect. Methods SetDrawFirstAndSecondAxis() and SetDrawThirdAxis() of the class Prs3d_DatumAspect have been corrected to ensure that myAxis may be set only to valid values of the enum, and avoid unsafe operations.</td>
</tr>
<tr>
<td>29184</td>
<td><strong>Summary:</strong> DrawWindow::Save() fails when using WinCodec with PNG codec. DrawWindow::Save() now uses Image_Format_BGR instead of Image_Format_BGR32 for better compatibility with image encoders when dumping WinAPI bitmap.</td>
</tr>
<tr>
<td>29225</td>
<td><strong>Summary:</strong> Font_FTFont::AdvanceX() retrieves kerning value for incorrect characters pair. A misuse of FT_Get_Kerning has been fixed within Font_FTFont::AdvanceX()/Font_FTFont::AdvanceY(). Font_FTFont::LoadGlyph() avoids returning TRUE if a method is called with 0 argument for a second time and more.</td>
</tr>
<tr>
<td>Line Numbers</td>
<td>Summary:</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>29262</td>
<td>AIS_InteractiveContext::Load() does not register Object in the Viewer.</td>
</tr>
<tr>
<td>29263</td>
<td>AIS_InteractiveContext::Load() and KeepTemporary() register object in the Viewer in the same way as Display() does.</td>
</tr>
<tr>
<td>29283</td>
<td>Summary: allow defining more than 8 light sources.</td>
</tr>
<tr>
<td>29284</td>
<td></td>
</tr>
<tr>
<td>29285</td>
<td></td>
</tr>
<tr>
<td>29290</td>
<td>Summary: TKOpenGl – allow defining Light source per Zlayer.</td>
</tr>
<tr>
<td>29291</td>
<td></td>
</tr>
<tr>
<td>29292</td>
<td></td>
</tr>
<tr>
<td>29293</td>
<td></td>
</tr>
<tr>
<td>29294</td>
<td>Summary: TKOpenGl – provide distance culling option.</td>
</tr>
<tr>
<td>29295</td>
<td></td>
</tr>
</tbody>
</table>
| 29300 | **Summary:** TKOpenGL – provide depth pre-pass option.  
OpenGl_LayerList::Render() now handles new option Graphic3d_RenderingParams::ToEnableDepthPrepass which prepends additional pass to rendering pipeline filling Depth Buffer in advance. |
| 29331 | **Summary:** TKOpenGL – make OpenGl_PrimitiveArray::IsFillDrawMode() as virtual method of OpenGl_Element.  
OpenGl_LayerList::Render() now handles new option Graphic3d_RenderingParams::ToEnableDepthPrepass which prepends additional pass to rendering pipeline filling Depth Buffer in advance. |
| 29337 | **Summary:** TKOpenGL – visual artifacts on Intel Broadwell GPU.  
Enable multiple draw buffers in shader program only if it is required by a specific application.  
occSetFragCol or () – a new GLSL function has been introduced as an alternative to setting occFragCol or / occFragCover age variables.  
TKOpenGL – uninitialized class field OpenGl_FrameBuffer::myIsOwnDepth has been fixed. |
| 29346 | **Summary:** TKOpenGL – collect frame statistics.  
The following tools collecting statistics internally have been implemented in TKOpenGL for performance analysis:  
• New option Graphic3d_RenderingParams::ToShowStats displays rendering statistics.  
• New class OpenGl_FrameStats accumulates frame statistics used for the currently rendered context.  
• OpenGl_View::Redraw() and OpenGl_View::RedrawImmediate() reset counters within OpenGl_Context::FrameStats().  
• OpenGl_Layer::UpdateCulling() simplifies resetting of culling state for cullable structures. |
| 29350 | **Summary:** OpenGl_Text – add Aspect_TODT_SHADOW text style.  
New style Aspect_TODT_SHADOW draws a tiny shadow at the right-bottom corner with one pixel shift, producing a much nicer visual look than Aspect_TODT_DEKALE. |
| 29366 | **Summary:** OpenGl_Text – artifacts when using Aspect_TODT_SHADOW Aspect_TODT_DEKALE at different zoom level.  
OpenGl_Text now applies Polygon Offset instead of Z-shift in world coordinates for drawing background.  
Set PolygonOffset() method has been moved from OpenGl_Workspace to OpenGl_Context. |
| 29372 | **Summary:** Graphic3d_TransformPers – improve description of Local Coordinate system defined by Transformation Persistence. |
| 29395 | **Summary:** V3d_Vi ew – Grid disappears forever after enabling Ray Tracing.  
Custom Graphic3d_Structure implementation has been added to V3d_ Angular Grid and V3d_Circular Grid to trigger recomputation in case of device lost. Primitive arrays are no more (re)computed while grid is not actually displayed. |
<table>
<thead>
<tr>
<th>Page 27</th>
</tr>
</thead>
</table>
| **Summary:** TKOpenGl – GLSL compilation errors on buggy OpenGL ES driver for PowerVR SGX 544MP  
OpenGl_ShaderProgram::Initialize() now defines THE_MAX_LIGHTS/ THE_MAX_CLIP_PLANES to zeros to provide a workaround for problems with buggy OpenGL drivers. |
| 29474 |
| **Summary:** TKOpenGl – MSAA FBO initialization failure on OpenGL ES 3.2 device.  
OpenGl_Vi ew::nyFboCol or For mat is now initialized using input texture format GL_RGBA8 on mobile platforms (as on a desktop platform). |
| 29477 |
| **Summary:** AIS_Shape – filter unsupported Display Modes within ::AcceptDisplayMode().  
AIS_Shape::AcceptDisplayMode() now accepts only modes 0,1 and 2. AIS_Col or edShape::Compute() no more computes presentation for unknown display mode. |
| 29491 |
| **Summary:** AIS_Point dynamic highlighting is not drawn on RedrawImmediate.  
Zlayer for Dynamic highlighting of AIS_Point has been set to Graphic3d_ZlayerId_Top and for Selected highlighting to Graphic3d_ZlayerId_UNKNOWN to follow the behavior of normal AIS object. |
| 29500 |
| **Summary:** TKOpenGl – fix access violation due to misprint in OpenGl_AspectMarker. |
| 29503 |
| **Summary:** TKOpenGl – visual artifacts on Adreno 305/308.  
OpenGl_ShaderManager now:  
• prefers GLSL ES 100 over GLSL ES 300 on devices reporting OpenGL ES 3.0;  
• prefers GLSL ES 300 on devices reporting OpenGL ES 3.1+.  
This provides a workaround for known buggy implementations of OpenGL ES 3.0 drivers. |
| 29508 |
| **Summary:** TKOpenGl – Weighted OIT + MSAA shader compilation errors.  
OpenGl_ShaderManager has been improved to fix implicit cast ivec2 -> vec2 and specify GLSL version to “320 es” on appropriate devices. |
| 29509 |
| **Summary:** Introduce Al phaMode property.  
New property AlphaMode from class Graphic3d_AlphaMode defines how Alpha value should be treated. |
| 29517 |
| **Summary:** TKOpenGl – fallback to Graphic3d_TOSM_FACET from Gouraud/Phong when nodal normals are undefined.  
Graphic3d_TOSM_VERTEX and Graphic3d_TOSM_FRAGMENT fall back to Graphic3d_TOSM_FACET instead of Graphic3d_TOSM_UNLIT when no normal attributes are defined for triangles array. As a result, Graphic3d_TOSM_UNLIT Shading Model or material should be specified explicitly without reflecting properties to preserve the old behavior. |
| 29519 |
Summary: Wrong validation of Anchor point for Radius Dimension.
The validation of Anchor point for Radius Dimension has been improved in method
AIS_RadiusDimension::IsValidAnchor.

Summary: TKV3d – User-defined texture coordinates corrupted by
StdPrs_ShadedShape.
The function fill Tri angles from StdPrs_ShadedShape.cxx has been protected
against void UV range of a face.

Summary: Add AI S_I nteractiveCont ext::Cl earDet ect ed() undoing
MveTo() dynamic highlighting.
New method AI S_I nteractiveCont ext::Cl earDet ect ed() allows resetting the
list of detected objects and clearing dynamically highlighted entity under the mouse
cursor after the previous AI S_I nter activeCont ext::MveTo().
Draw command vmoveto has been extended with new argument - reset.

Summary: SelectMgr_Vi ewerSel ect or::Pi ck edPoi nt() returns a wrong 3D
point for objects with transformation persistence.

Summary: TKOpenGl - structure is entirely clipped by suppressed clipping.
The check for disabled state of the plane has been added in
OpenGl_Structure::Render().
Draw command vclipplane has been extended with new option -
set OVERRIDEGLOBAL.

Summary: Wrong result of SelectMgr_RectangularFrustum::IsClipped.
SelectMgr_RectangularFrustum::computeClippingRange() does not skip
depth range starting behind the ray.

Data Exchange
Summary: Segfault when transferring HLR-created shapes via STEPCont rol_ W r iter.
Protection against null shape on writing has been added in XSControl_ W r kSessi on.
The status I FSel ect _ Ret Voi d will be returned in such case (instead of access
violation).

Summary: Exception is raised with no result during reading file 2033zsh1_1.stp.
Static function stepstrcmp() from StepData_StepReaderData.cxx has been
replaced by plain strcmp() to compare the full strings (stepstrcmp() returned true if
one of the strings was longer than the other but the common part was equal).
Protection has been added to avoid exception for cases when representation entities are
NULL.
| 29119 | **Summary:** Documentation for PMI in XCAF.  
Documentation for GD&T, Clipping planes and Saved view components has been added to the XDE User's Guide. |
| 29282 | **Summary:** UpdateAssemblies is not working for located root assemblies.  
XCAFDoc_ShapeTool has been modified to check for root assemblies having their own location (when free shape is an instance for main assembly). |
| 29338 | **Summary:** Add Planes for Tolerance zones in Geometric tolerances.  
The possibility to store Tolerance zones defined by orientation or intersection planes has been added in XCAF as XCAFDim Tol Object s_ToleranceZoneAffectedPlane. |
| 29362 | **Summary:** Crash during reading step file.  
Additional check for wires has been added in STEPCAFControl_Reader. |
| 29391 | **Summary:** Invalid import of TrimmedSurface.  
Import of TrimmedSurface has been fixed in IGSToBRep_TopoSurface. |
| 29403 | **Summary:** Subshape names are not imported from STEP.  
The STEP processing of subshape names has been fixed in STEPCAFControl_Reader and STEPCAFControl_Writer. Broken subshape creation in Document as tree has been replaced with plain subshapes structure. |
| 29436 | **Summary:** Extend "Expand compounds" functionality.  
XCAFDoc_ShapeTool has been modified to expand (convert from a part to assembly) not only compounds, but other container shape types: compsolid, shell and wire. |
| 29525 | **Summary:** PMI dimension names.  
Semantic PMI names translation from STEP to XCAF has been implemented in OCCT: |
| 29526 | **Summary:** Test Harness command ReadIges does not support read.iges.onlyvisible mode.  
DRAW command ReadIges now takes into account the current setting of the parameter read.iges.onlyvisible. |
| 29597 | **Summary:** Unable to read VRML2 file.  
VrmlData_Scene::createNode() now handles Collision item.  
VrmlData_ArrayVec3d::ReadArray() now handles a case with omitted brackets. |
<table>
<thead>
<tr>
<th>Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Possible exception in shape tool.</td>
<td>Expand compound operation has been fixed to properly calculate location of subshapes and provide correct sharing for new parts. Auto-naming has been switched-off for this operation.</td>
</tr>
<tr>
<td>Access violation in StepVisual_PresentationStyleAssignment</td>
<td>Several checks for null have been added in STEP translation procedure.</td>
</tr>
<tr>
<td>Draw Harness – reshape command usage is not properly documented.</td>
<td>The implementation of reshape command has been corrected according to OCCT coding rules.</td>
</tr>
<tr>
<td>Exception while projection 2D-point on 2D-line.</td>
<td>DRAw command 2dproj now can return not only 2D line, but also 2D point as extremum.</td>
</tr>
<tr>
<td>DBRep_DrawableShape – fix inappropriate use of unordered map.</td>
<td>TColStd_DataMapOfIntegerInteger has been replaced by NCollection_Vector in DBRep_DrawableShape.cxx.</td>
</tr>
<tr>
<td>Unclear syntax of add command.</td>
<td>Help message for add command has been corrected.</td>
</tr>
<tr>
<td>Uniform mechanism providing History of shape modifications for OCCT algorithms in DRAw.</td>
<td>The mechanism for unification of history commands has been implemented for all OCCT algorithms.</td>
</tr>
<tr>
<td>vtexture command crashes.</td>
<td>Null check has been added for &quot;off&quot; option of vtextur e command to avoid access violation.</td>
</tr>
</tbody>
</table>

**Draw**

The following Draw commands should be used to track the history of shape modifications of any operation:

- modified finds the shapes modified from the given shape in the given history;
- generated finds the shapes generated from the given shape in the given history;
- isdeleted checks if the given shape has been deleted during operation.

The mechanism allows fast and easy enabling of the DRAw history support for the algorithms supporting the history on the API level (i.e. the algorithm should have methods Modified(), Generated() and IsDeleted()).
| 29735 | **Summary:** Command to set 2D mode for viewer in ViewerTest package. 
Draw command `vinit` has been extended with new option `-2d_mode`. 
New command `v2dmode` switching on/off the mode has been added. |
| 29739 | **Summary:** Command `vdonly` does not hide displayed objects. |
| 29784 | **Summary:** Crash at STEP file reading with enabled sub-shapes. 
Method `STEPCAFControl_Reader::ExpandShell` has been protected against the case when `Connected_Face_Set` entity contains not only `FACE` entities. |

**Mesh**

| 29229 | **Summary:** Crash at `Poly_Triangulation::Normal`. 
Creation of `gp_Dir` has been fixed in `Poly_Triangulation.cxx`. |
| 29715 | **Summary:** Estimate the grid size of the acceleration structure by the complexity of the face. 
`BRepMesh_Delaun` algorithm uses a grid as an acceleration structure for finding the circles and triangles, which can contain a point. Now the size of this grid is estimated basing on the complexity of the face and the desired face deflection. |

**Samples**

| 27736 | **Summary:** Rectangle selection issues within MFC sample Viewer3d. 
`Cvi ewer3dVi ew Cani mati onVi ew3D` and `COCCDenVi ew now use Al S_Rubber Band`. |
| 29069 | **Summary:** Handle `UNICODE` filenames within C++/CLI Csharp sample. |
| 29083 | **Summary:** Specify multiple Make jobs within `make.sh` for Qt sample. |
| 29140 | **Summary:** Viewer is not updated in MFC Modeling and Viewer3D samples. 
OCC Viewer is now automatically updated after erasing objects, operation Common and drawing sphere. |
| 29393 | **Summary:** Androi dQt sample build fails. 
Androi dQt sample has been updated to take into account latest changes in OCCT: 
- Obsolete arguments in `Androi d Qt` are now avoided within `V3d_Vi ew.Set W ndow()` usage. 
- Missing `NativeFBConfig()` method declared in the interface has been added in `Androi d Qt_W ndow`. |
| 29394 | **Summary:** IESample contains strange STEP reading code. 
The order of operations in `Translate::importSTEP` procedure has been fixed: all STEP roots are transferred first and then all shapes are read. The resulting sequence is not discarded and recreated. |
| 29470 | **Summary:** Eliminate references to deprecated Local Context from MFC sample.  
Methods `MoveTo()`, `Select()` and `ShiftSelect()` from `AIS_InteractiveContext` now throw an exception on invalid `V3d_View` argument instead of returning empty results.  
`AIS_InteractiveContext::DetectedShape()` and `BeginImmediateDraw()` can be called without opened Local Context.  
Unused Draw commands `vsetam` and `vunsetam` as well as methods `StandardModeActivation()`, `PickObject()` and `PickObjects()` from `ViewerTest` have been removed.  
Interactive input of Selection modes 0..7 now redirects to `vsetmode` instead of removed `ViewerTest::StandardModeActivation()`.

| 29559 | **Summary:** Wrong copyright statement in `FuncDemo`.  
`FuncDemo` sample has been updated according to new `elasticnodes` example in Qt.

| 29571 29668 | **Summary:** Build Qt samples together with OCCT.  
`BUILD_MODULE_QtSamples` flag has been provided in CMake to switch ON/OFF the compilation of Qt samples.

| 29631 29643 | **Summary:** Build Android sample together with OCCT on Windows platform.  
The environment for building Android sample on Windows platform has been created.  
WIN32 definition now provides functionality for sample on Windows.  
In CMake procedure, `Qt5_FOUND` variable is now defined before compilation of modules. Search of Qt packages has been moved into `qt.cmake`.

| 29659 | **Summary:** Image is not displayed in V3d MFC sample.  
Wrong Display Mode assigned to `Sample2D_Image` presentation has been fixed.

| 29674 29748 29800 | **Summary:** Improvements in Inspector tool.  
The following improvements have been implemented in Inspector tool:  
- Preferences now store user-defined state of positions and visibility of dock widgets, visibility and width of tree view columns, 3D view projection and the folder containing recently opened files;  
- New V3d_Cont role package unites common functionality shared between different plugins;  
- New “Export to ShapeView” functionality processes Location and Orientation for external TopoDS_Shape object;  
- F5 key can be used to update the content of each plugin;  
- New “Visibility” column is available in the tree view;  
- New “Properties” tree view item presents a tree of current Filters of context.
<table>
<thead>
<tr>
<th>Case Number</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>29733</td>
<td>Inspector tool - crash after selecting TNaming_usedShapes tree item.</td>
</tr>
<tr>
<td>29743</td>
<td>- TNaming_usedShapes now avoids calling methods of an empty TopoDS_Shape;</td>
</tr>
<tr>
<td></td>
<td>- TNamingNamedShape avoids NULL TNaming_UsedShape if TopoDS_Shape is NULL(TNaming_Builder::Delete).</td>
</tr>
<tr>
<td></td>
<td>- DFBrowser Pane presentation tree item has been corrected</td>
</tr>
<tr>
<td></td>
<td>-Obsolete methods of processing SortedReferences have been removed.</td>
</tr>
<tr>
<td>29741</td>
<td>Inspector tool - wide icon size in the table of TNamingNamedShape attributes in DFBrowser plugin.</td>
</tr>
<tr>
<td>29747</td>
<td>Inspector tool - start DFBrowser inside FuncDemo Qt sample.</td>
</tr>
<tr>
<td></td>
<td>New Model-&gt;DFBrowser action starts Inspector tool with active DFBrowser plugin filled by the sample OCAF application.</td>
</tr>
<tr>
<td>29749</td>
<td>Inspector tool - Remove &quot;modified&quot; column from TNamingNamedShape presentation.</td>
</tr>
<tr>
<td>29781</td>
<td>Inspector tool - history tree model is not set into external callback in VInspector.</td>
</tr>
<tr>
<td></td>
<td>The following modifications have been introduced to fill a VInspector_CallBack object created in a custom application by the VInspector history model:</td>
</tr>
<tr>
<td></td>
<td>- AIS_InteractiveContext and VInspector_Vi ewModel History are set into VInspector_CallBack;</td>
</tr>
<tr>
<td></td>
<td>- displaySelectedPresentations() has been corrected to Hide/Show (by popup menu) any AIS_Inter activeObject, not only AIS_Shape (e.g. AIS_Tri hedron).</td>
</tr>
</tbody>
</table>

**Documentation**

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>28660</td>
<td>Describe how to fulfill LGPL terms in OCCT-based applications.</td>
</tr>
<tr>
<td></td>
<td>It is now described in the Overview how to fulfill LGPL requirements when OCCT is used in proprietary applications.</td>
</tr>
<tr>
<td>29513</td>
<td>Replace OCC logo in user guides.</td>
</tr>
<tr>
<td></td>
<td>OCC user guides now show OCC logo in higher quality.</td>
</tr>
<tr>
<td>29545</td>
<td>VIsualization.md - article incorrectly specifies that AIS_Connect editnteractive view can define own Material.</td>
</tr>
<tr>
<td>29730</td>
<td>OCAF User Guide updated for the case of attributes with a user-defined GUID.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Issue Number</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>22651</td>
<td>Impossible to build OCC as a static library due to using Standard_EXPORT instead of Standard_API. All library-specific macros for defining export / import properties of symbols on Windows (such as St andar d API, _Draw_API, _math_API, etc.) have been eliminated. Common macro St andar d EXPORT is used in all places where it is necessary.</td>
</tr>
<tr>
<td>28090</td>
<td><strong>Summary:</strong> DRAWHOME environment variable missing in CMake. DRAWHOME variable has been replaced by CSF OCCTResour cePat h to run successfully DFBr owse command in Draw launched from Visual Studio. Missing icons of folders, attributes and named shapes used in DFOpen1 rage command have been added in src\DrawResour ces\dftree.tcl.</td>
</tr>
<tr>
<td>28335</td>
<td><strong>Summary:</strong> CM ake – 3rd-party library names present in two places and are not synchronized with each other. Library names from file adm\cm ake/occt_csf.cma ke are now used to search for libraries. Hardcoded variants of tcl/tk library names used for searching have been removed.</td>
</tr>
<tr>
<td>28971</td>
<td><strong>Summary:</strong> Problem compiling OCCT 7.2 with glibc 2.26. St andar d_Cl ocal eSent ry does not include xl ocal e. h when using glibc anymore. HAVE_XLOCAL E_H has been renamed to OCCT_CLOCALE_POSIX X2008 to avoid confusion. Macros OCC_CHECK_BASE_CLASS has been renamed into OCCT_CHECK_BASE_CLASS.</td>
</tr>
<tr>
<td>29075</td>
<td><strong>Summary:</strong> Fix TKService linkage errors due to usage of GLX functions while using EGL. The use of GLX functions to choose Visual when building with HAVE_EGL/HAVE_GLES2 options has been fixed in Xw_W ndow.</td>
</tr>
<tr>
<td>29112</td>
<td><strong>Summary:</strong> Image_VideoRecorder – compilation fails on Ubuntu with Li bavutil 54.x (FFmpeg 2.7.6) The macro Pixel_xe! For m at has been undefined to allow compilation with Li bavutil 54. x. Coding rules on naming of classes have been revised to describe the correspondence of names of public types and files.</td>
</tr>
<tr>
<td>29118</td>
<td><strong>Summary:</strong> Incorrect generation of reference documentation for modules. Generation of reference documentation for single modules has been corrected to provide dependencies between modules in the graph.</td>
</tr>
<tr>
<td>29129</td>
<td><strong>Summary:</strong> Incomplete support of MSVS2017. MSVC_VERSI ON is now used in CMake scripts instead of MSVC10, MSVC11, MSVC12, etc.</td>
</tr>
</tbody>
</table>
| 29169 | **Summary:** Fix compilation with undefined UNICODE on Windows.

Use of TEXT macros and of OSVERSIONINFO instead of OSVERSIONINFOW has been eliminated.

LoadIcon and LoadCursor are now used instead of LoadIconW and LoadCursorW when passing macros to standard resources (which depend on UNICODE flag). |
| --- | --- |
| 29249 | **Summary:** Standard_Failure compilation fails on VS2013 + Intel Compiler due to unavailability of thread_local.

INTEL_COMPILER version is now checked in combination with _MSC_VER on Windows. |
| 29250 | **Summary:** TKIVtk – build failure with VTK 6.3+ due to removal of vtkRenderi ngFr eeTypeOpenGL.

vtkRenderi ngFr eeTypeOpenGL has been removed from EXTERNLIB if vtk version 6.3.0 and above is used. |
| 29255 | **Summary:** .gitignore – do not track generated files with extensions VC.db and VC.opendb.

The files *.VC.opendb and *.VC.db generated by Visual Studio are not tracked by Git repository. |
| 29266 | **Summary:** CMake install does not copy PDB files in Debug mode.

The variable OCCT_INSTALL_BIN_LETTER is now defined only if the compiler is a version of Microsoft Visual C. |
| 29277 | **Summary:** Mingw-w64 build fails for TKOpenGL due to missing link to OpenGL.

The order of external libraries has been corrected in TKOpenGL / EXTERNLIB to mention high-level libraries earlier than low-level libraries, on which the former depend (e.g. Gl2Ps before OpenGL) and thus ensure that GCC linker can resolve dependencies. |
| 29297 | **Summary:** CMake – CSF_d3d9 should be processed for building TKD3Dhost using MinGW. |
| 29398 | **Summary:** List lex and yacc files in the StepFile/FILES to avoid CMake warnings.

Step.lex and step.yacc have been added to StepFile/FILES. |
| 29377 | **Summary:** CMake – linkage errors while using static OCCT libraries on Windows platform.

Macro OCCT_STATI C_BUILD is now defined when building OCCT as Static libraries. |
| 29398 | **Summary:** List lex and yacc files in the StepFile/FILES to avoid CMake warnings.

Step lex and step yacc have been added to StepFile/FILES. |
| 29407 | **Summary:** Allow MFC samples to be built when OCCT is linked statically.

CMake scripts have been corrected to perform search of libs and dlls of third-party libraries even for static builds of OCCT. The processing of errors in this case is relaxed: non-found DLLs are reported as warnings, and not found libs as warnings for a static build, and continuable errors for a shared build. |
The code of samples allows building with OCCT linked statically. The implementation of the main application class now ensures that initialization is done after creation of all global objects. The recommendations about generation of static libraries have been added in the building guide.

Summary: CMake – OpenGL ES should be available for Linux.

It is now possible to optionally build OCCT with GLESv2 and EGL on Linux.

Summary: Upgrade.dat – include deprecated enums into section [rename].

V3d_TypeOfShadingModel -> Graphic3d_TypeOfShadingModel and V3d_TypeOfLight -> Graphic3d_TypeOfLightSource renames have been added in upgrade.dat.

Summary: CMake – rename options for building samples.

- BUILD_SAMPLES_MFC is now used instead of BUILD_MODULE_MfcSample;
- BUILD_SAMPLES_QT is now used instead of BUILD_MODULE_QtSample;
- Sub-folder samples/mfc/ now groups MFC samples;
- Qt processing of *.ts resource files avoids creating excessive projects under Samples folder in VStudio.

Summary: Modification of a standalone build of Inspector tool.

It is now possible to compile Inspector in a Standalone mode to use this tool for earlier versions of OCCT. Correspondingly:
- The folder samples/tool/s/Tstandalone has been removed. CMake should use tools/CMakeLists.txt filename to build Inspector out of OCCT;
- The folder TInspectorEXE has been moved from samples/tools/ to tools/;
- TInspectorAPI_Version.hxx provides compilation of Inspector with earlier versions of OCCT.

Coding

Summary: ICC compiler warnings on Windows.

NCollection_UtfString and NCollection_UtfIterator classes have been refactored to use methods overloading instead of switches to dispatch implementation depending on character (Unicode code unit) size.

ICC-specific preprocessor directives have been added to avoid warnings.

Unused local functions and variables, class methods, unreachable statements, and extra throw() declarations reported by ICC have been removed.

Usage of expl for the name of local variable is avoided as it conflicts with standard C function expl defined in math.h as preprocessor macro.

Non-standard (MS-specific) argument envp has been removed from definition of main() function on Windows. Functions _main_ and _WinMain_ have been renamed to Draw_Main and Draw_WinMain, respectively, to avoid using names reserved in C++.
<table>
<thead>
<tr>
<th>Summary</th>
<th>Compiled Products for Android on Windows and Linux platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The compilation of Products for Android has been integrated into Windows and Linux platforms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Eliminate compiler warnings in OCCT samples.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Qt warnings about compilation problems under MSVC 2013 and greater have been eliminated. Order of includes has been changed to avoid warning about M_PI, redefinition warning of nat h. h, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Eliminate GCC compiler warnings -Wimplicit-fallthrough.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New macro Standard_FALLTHROUGH has been defined for use in a switch statement immediately before a case label, if the code associated with the previous case label may fall through to that next label (i.e. does not end with &quot;break&quot; or &quot;return&quot; etc.). This macro indicates that the fallthrough is intentional and should not be diagnosed by a compiler that warns on fallthrough.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Eliminate GCC compiler warnings -Wmissing-indentation when using MinGW.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Eliminate deprecation compiler warnings when targeting MacOS 10.12.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>AppleCLang 9 compiler warning &quot;binding dereferenced null pointer to reference has undefined behavior&quot;. Returning reference to null in AppDef_MyLineTool is now avoided.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>GCC 7.1 compiler warnings -Wmaybe-uninitialized on gp_XYZ and siblings.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The code has been corrected to avoid local variables of reference type pointing to fields of temporary objects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>GCC 7.1 warnings -Wstrict-aliasing in Graphic3d_ArrayOfPrimitives.hxx.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method Graphic3d_ArrayOfPrimitives::SetVertexColor() accepting color as three double RGB values and Graphic3d_Vec4ub object have been refactored to avoid using reinterpret_cast between pointers to complex types. A similar correction has been made in a static function VdrawSphere from ViewerTest_ObjectCommands.cxx.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Define rule for avoiding header inclusion list pollution.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It is now required that the source or header file should include only a minimal set of headers necessary for compilation, without duplicates (considering nested includes).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Eliminate GCC compiler warnings -W or nat - overflow.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSD_DirectoryIterator and OSD_FileIterator now use TCollection_AsciiString instead of unsafe 37print.</td>
</tr>
</tbody>
</table>
Summary: Remove Graphic3d_Vector or duplicating gp_XYZ.

Graphic3d_Vector class has been replaced by classes gp_Pnt / gp_XYZ / gp_Dir depending on context.

Unsafe float math causing out-of-range color results has been fixed in method StdSelect_ViewerSelector3d::ToPixMap().

Summary: Multiple compiler warnings in Inspectors.

The following modifications fix compiler warnings in Inspectors:
- Specific header files Standard_WarningsDisable.hxx and Standard_WarningsRestore.hxx have been added to disable and restore compiler warnings (currently only MSVC compiler is handled).
- Compiler warnings have been disabled for all includes of Qt headers.
- Warnings caused by floating point values in integer calculations are avoided.
- Use of CMAKE_AUTOMOC is avoided, header files are collected with Q_OBJECT iterating through project files.

Summary: TColStd_PackedMapOfInteger – declare Iterator as nested class of map collection.

Summary: Make V3d_Viewer::PrivilegedPlane() return const reference rather than a temp object.

Summary: NCollection_IndexedDataMap – add missing documentation to method Add().

Summary: IntWalk_Pwalking::PutToBoundary(...) method results in appearing several coincident points in Walking-line.

Check for coincident points has been added in intWalk_Pwalking algorithm.

Summary: Remove unused declarations from package Aspect.

Unused declarations Aspect_TypeOfPrimitive, Aspect_TypeOfLayer, Aspect_TypeOfEdge, Aspect_TypeOfConstraint, Aspect_DriverDefinitionError and Aspect_BadAccess have been removed.

Summary: Bnd_Range – inconsistent methods pair GetMin() / GetMax().

The following modifications have been introduced in class Bnd_Range:
- Method GetMAX() has been renamed to GetMax().
- New method Add() takes another Bnd_Range as argument.
- New methods IsOut() mimics a Bnd_Box interface.
- Methods Shift() and Shifted() no more modify Void range.

Summary: Avoid usage of Standard_EXPORT attribute for inline methods.

All occurrences of Standard_EXPORT attached to inline methods in OCCT code have been eliminated. Some unused classes and C++ files producing no code have been deleted.

Summary: Replace Standard_Integer with Graphic3d_ZLayerId for consistency.

Standard_Integer has been replaced with Graphic3d_ZLayerId in methods AIS_InteractiveContext::SetZLayer and PrsMgr_Presentation::SetZLayer.
## 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, ARM64 and x86), and iOS (ARM64) 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 [https://www.opencascade.com/content/system-requirements](https://www.opencascade.com/content/system-requirements).

<table>
<thead>
<tr>
<th><strong>Supported Platforms</strong></th>
<th><strong>Pre-requisites</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux Operating System</strong></td>
<td>Arch Linux, CentOS 6.4, CentOS 7.3, Fedora 22, Fedora 24, Ubuntu 1604, Debian 7.0, Debian 8.0</td>
</tr>
<tr>
<td><strong>Windows Operating System</strong></td>
<td>MS Windows 10 / 8 / 7 SP1 / Vista SP2 / XP SP3</td>
</tr>
<tr>
<td><strong>OS X/macOS Operating System</strong></td>
<td>OS X/macOS 10.10 and later</td>
</tr>
<tr>
<td><strong>Android Operating System</strong></td>
<td>Android 4.2 and above</td>
</tr>
<tr>
<td><strong>iOS Operating System</strong></td>
<td>iOS 7 and above</td>
</tr>
<tr>
<td><strong>Minimum memory</strong></td>
<td>512 MB, 1 GB recommended</td>
</tr>
<tr>
<td><strong>Free disk space</strong> (complete installation)</td>
<td>650 MB of disk space, or 1.4 GB if installed with reference documentation</td>
</tr>
<tr>
<td><strong>Graphic library</strong></td>
<td>OpenGL 3.3+, OpenGL ES 2.0+</td>
</tr>
</tbody>
</table>

### C++

**For Linux:**
- GNU gcc 4.3+
- LLVM Clang 3+

**For Windows:**
- Microsoft Visual Studio 2008
- Microsoft Visual Studio 2010 SP1
- Microsoft Visual Studio 2012 Update 4
- Microsoft Visual Studio 2013 Update 2
- Microsoft Visual Studio 2015
- Microsoft Visual Studio 2017
- Intel C++ Composer XE 2013 SP1
- GCC 4.3+ (Mingw-w64)

**For Mac OS X:**
- XCode 6 or newer

**For Android:**
- GCC 4.8+ (android-ndk-r12+)

### TCL (for testing tools)

**For Linux:**
- Tk 8.6.3+ [https://www.tcl.tk/software/tcltk/8.6.html](https://www.tcl.tk/software/tcltk/8.6.html)

**For Windows:**
- Tk 8.6.3+ [https://www.tcl.tk/software/tcltk/8.6.html](https://www.tcl.tk/software/tcltk/8.6.html) or ActiveTcl 8.6 [https://www.activestate.com/activetcl/downloads](https://www.activestate.com/activetcl/downloads)

**For OS X:**
- Built-in Tk/Tk 8.6+

### Qt (for demonstration tools)

- Qt 4.8.6 [https://download.qt.io/](https://download.qt.io/)

### FreeType (OCCT Text rendering)

- FreeType 2.4.11-2.7.1 [https://www.freetype.org/](https://www.freetype.org/)

### FreeImage (Support of common graphic formats)


### gl2ps (Export of OCCT viewer contents to vector graphic file, deprecated)

- gl2ps-1.3.8 [http://geuz.org/gl2ps/](http://geuz.org/gl2ps/)

### TBB (optional tool for multithreaded algorithms)

- TBB 4.x or 5.x [https://www.threadingbuildingblocks.org/](https://www.threadingbuildingblocks.org/)

### Doxygen (optional for building documentation)

- Doxygen 1.8.5+ [https://www.stack.nl/~dimitri/doxygen/download.html](https://www.stack.nl/~dimitri/doxygen/download.html)

### FFmpeg (multimedia framework for OCCT video recording)

- ffmpeg-3.3 [https://www.ffmpeg.org](https://www.ffmpeg.org)