Open CASCADE Technology and Products ver. 6.8.0

Release Notes

Overview

Open CASCADE Technology and Products version 6.8.0 is a minor release, which includes nearly 600 improvements and bug fixes over the previous release 6.7.1.

Version 6.8.0 is binary incompatible with the previous versions of Open CASCADE Technology and Products, so applications linked against a previous version must be recompiled to run with this Version 6.8.0.

Highlights

Foundation Classes:

- **Using `NULL` as invalid pointer in Handle classes, instead of custom constant (0xfefd0000...);**
- **STL-compatible iterators for classes in `NCollection` package;**
- **Code clean-up: removing usage of `config.h`, support of pre-standard STL streams, unused CDL template ("generic") classes and instantiations, etc.;**

Modelling:

- **Data structures for Bounded Volume Hierarchy (BVH) algorithms;**
- **Two new algorithms of global optimization in math, used for Extrema;**
- **Parallelization of Boolean Operations algorithm;**
- **Interface to break execution of Boolean Operations algorithm;**
- **Possibility to protect shape against modification of geometry;**
- **Refactoring and optimization of `BRepMesh` algorithm;**
- **New option in 2d offset algorithm allowing to keep sharp corners and build one-side offset on open wire;**

Visualization:

- **New component, VIS, providing interactive services (similar to AIS) for OCCT shapes in VTK viewer;**
- **New approach for manipulations with views using camera paradigm;**

- Support of stereoscopic display (requires graphic card supporting OpenGL Quad Buffer);
- Improved support of perspective views;
- Ray tracing now uses shaders (GLSL) instead of OpenCL;
- Possibility to combine objects drawn by OpenGL and ray tracing in one view;
- Frustum culling for fast display of large number of objects with high zoom;
- Initial support for OpenGL ES 2.0 for mobile platforms;
- New classes for display of colored shapes and point clouds in AIS;
- Revised and completed implementation of connected interactive objects;
- Improved handling of temporary objects in the viewer (immediate mode);

Data Exchange:

- Support of COMPSOLIDs in STEP export;
- Support of UNICODE (UTF-8) filenames;
- Support of names and attributes assigned to points (vertices) in XCAF;

DRAW:

- FPE signal handlers are disabled by default;
- New sample scripts demonstrating modeling and visualization capabilities in DRAW;
- Improved usability of top level menu (help browser, samples, User Guide);

Documentation:

- New User Guide describing Boolean Operations algorithms;
- Improvement of extraction of class documentation for Reference Manual;

Samples:

- New sample for using drawing OCCT 3D viewer into Direct3D surface in WPF applications;

Build system:

- Support of building on Android (except for DRAW);

Products:

- New (experimental) mode in Express Mesh, allowing generation of all-quad meshes;
- Universal interface to export mesh produced by Express Mesh;
- Improved triangulation near face boundaries in Express Mesh (Delaunay meshing);
- Update of training materials and samples;
- Translation of HATCH entities from DXF;
- Boolean operations on meshes in OMF provide history on elements and handle complex cases;
- Visualization of meshes in OMF takes into account direction of normal vectors to surface;
- New wrapper for OCCT classes for Java language;

Table of Contents

Modifications

Foundation Classes

22125 22484 24716 25367 25369	*Summary:* UNICODE characters support. The behavior of all functions that accept `Standard_CString` at input has been changed, so that now the strings are assumed to be in UTF-8 encoding. File functions taking wide characters wchar_t (e.g. `_wopen`) are now used on Windows platform, using `TCollection_ExtendedString` for UTF-8 to UTF-16 conversion.
24342	*Summary:* TKernel - Create methods for direct access to coordinate components. New methods `gp_XY::ChangeCoord` and `gp_XYZ::ChangeCoord` have been implemented for direct access to coordinate components.
24405	*Summary:* Implement aligned allocator. New class `Ncollection_AlignedAllocator`, methods `Standard::AllocateAligned()` and `Standard::FreeAligned()` as well as macros `STANDARD_ALIGNED` have been added to implement aligned allocator using platform-specific APIs for SSE optimization.
24533	*Summary:* Use 0 to check null handle instead of `UndefinedHandleAccess`. Handle classes now use 0 as invalid value for pointer instead of a custom and platform-dependent value. Compiler macros `UndefinedHandleAddress` and `_OCC64` have been eliminated.
24473 24669 25059 25164 25411	*Summary:* `TKMath, BVH` – introduce template-based package for Bounding volume hierarchy structures and tools. New template-based package `TKMath/BVH` provides tools and algorithms using Bounding Volume Hierarchy methods. A bounding volume hierarchy (BVH) is a tree structure on a set of geometric objects. All geometric objects are wrapped in bounding volumes (`BVH` package uses AABB - axis-aligned bounding boxes) that are organized into a binary tree. This accelerating structure can be used in a wide variety of geometry and graphics algorithms, such as ray-tracing, projecting, collision detection, search of nearest neighbor, frustum culling, efficient selection and others. `BVH` package includes a variety of BVH-building methods, which provide various approaches to the choice between construction time and resulted tree quality.
24831	*Summary:* Make iterators of `NCollection` classes STL-compatible. The following modifications have been introduced in frame of this issue: ▪ STL-compatible iterators returned by methods `begin()` and `end()` have been provided in classes from `NCollection` package. ▪ `NCollection_Array1::Iterator` has been redesigned to use pointer instead of index. ▪ Iterators of `Sequence, Array`, and `Vector` have been extended by new methods to iterate backwards.

24911	*Summary:* Avoid using virtual functions in `NCollection` classes. The following modifications have been introduced in frame of this issue: ▪ The class `NCollection_BaseCollection` and macro `DEFINE_BASECOLLECTION` have been removed. It means that methods `Assign()` from other compatible collections (via inheritance of `BaseCollection`) or base Iterator class are not available any more. ▪ All methods of Iterator classes are made non-virtual, allowing their inline expansion for better performance. ▪ OCCT-specific operators new and delete have been added to collection classes and removed from iterator classes.
24971	*Summary:* Incomplete interface of `NCollection` classes. `NCollection` classes have become compatible with `TCollection` equivalents. The following improvements have been implemented: ▪ Copy constructor in List and Maps is now used for placement of new items in collection instead of assignment operator, thus default constructor is not necessary anymore for the item class ▪ Constructors with additional argument of element type have been added in array classes operated by `Handle` and defined by `NCollection_DefineHArray*.hxx`, which allows initializing an array immediately by the specified value ▪ The following methods have been added: non-constant methods `List::First()` and `List::Last()`; `TListIterator::Value()`; `NCollection_DefineHSequence::Append()` accepting `Handle(HSequence)`; ▪ Default implementation of global function `IsEqual()` is provided as template; ▪ The code using lists and maps of sequences has been refactored to operate sequence by `Handle` (since `Sequence` does not need to have a public copy constructor). ▪ In addition, error checking code has been simplified to use macros `_Raise_if` instead of custom `#ifdefs` with the same meaning. ▪ Comments within declaration of instances of generic classes have been removed in CDL. ▪ The bug corrupting data if the original vector is empty has been fixed in copy constructor of `NCollection_BaseVector`.
25064	*Summary:* `TCollection` modification for Salome porting. Private copy constructors have been replaced by public and valid copy constructors in some classes from `TCollection` package. These classes now can be used in other collections that use copy constructors including third party ones.
24834	*Summary:* Memory allocation for an exception message must not throw another exception. Functions `allocate_message` and `deallocate_message` from `Standard_Failure` class now use `malloc/free` instead of operator `new/delete` to avoid `bad_alloc` exception.

24897	*Summary:* Uncontrolled exit if scanner fails in `ExprIntrp`. Macro `YY_FATAL_ERROR()` has been introduced in `ExprIntrp.lex` to raise an exception instead of program exit in case of scanner error. Fixed-length string buffers have been replaced by `TCollection_AsciiString` to avoid buffer overflow.
24908	*Summary:* Make `OSD_MallocHook` work under MSVS 2012. Version check has been corrected to support versions greater than VS2008. A dummy instruction has been inserted in method `OSD_MAllocHook::Callback` so that the compiler could leave place for a break point.
24931	*Summary:* Stack overflow when writing large shapes to XML. The code of `LDOM_OSStream` class has been refactored to use `NCollection_IncAllocator`, which avoids the destructor and improves the performance.
24998	*Summary:* Incorrect `USE_MATH_DEFINES` definition in `Standard_math.hxx` prevents building client applications. The clause `#ifndef` has been added in class `Standard_math` to avoid compilation error.
25000	*Summary:* Missing method implementation. Method `NCollection_SparseArrayBase::changeValue()` has been replaced by `NCollection_SparseArray::ChangeValue()`.
25005	*Summary:* Global function `HashCodes()` is not exported on Windows. Declaration of global function `HashCodes()` in `Standard_CString.hxx` has been corrected.
25024	*Summary:* `BSplCLib::PrepareInsertKnots` reports incorrect number of poles. The problem with incorrect number of poles returned by method `BSplCLib::PrepareInsertKnots` has been fixed.
25057	*Summary:* `Message_Algorithm` fails to find messages defined for the base class. New method `HasMsg()` has been added in `Message_MsgFile` class to check if a message with a given key is registered. That method is used in `Message_Algorithm` to check if a message is defined on a given level of class hierarchy. Error message generation in `Message_MsgFile::Msg()` has also been revised: now it includes a message key and gets added to the registry, to avoid re-generation in case of multiple requests. Access to message registry maintained by the `Message_MsgFile` class is protected from concurrent access by mutex.

25101	*Summary:* Typo in code documentation of `gp_Trsf::SetValues.` A typo in code documentation of `gp_Trsf::SetValues` has been fixed.
25252	*Summary:* Slowdown in reading of `.brep` on VS2011. New function `GeomTools::GetReal` has been implemented to read reals from strings using Strtod() function instead of C++ stream interface, providing accelerated reading of BRep files.
25253	*Summary:* `gp_trsf` - code revision. The definition of transformation in method `gp_Trsf::SetTranslationPart` has been corrected.
25257	*Summary:* Expression should use string names for comparison. The classes `Expr_NamedExpression` and `Expr_NamedUnknown` have been modified in to use string names for comparison.
25261	*Summary:* `NCollection` extension. Two new allocators have been implemented in `NCollection` package: ▪ `WinHeapAllocator` creates a dedicated heap for allocations. It may be used to take control over memory fragmentation because on destruction ALL allocated memory will be released to the system. ▪ `AccAllocator` allocates memory on request returning the pointer to the allocated space. The allocation units are grouped into blocks requested from the system. This memory is returned to the system when all allocations in a block are freed. This method is faster than the standard `new()` and `malloc()` calls and consumes very small additional memory to maintain the heap. Additionally, access in data map has been optimized, so that seek and find functions return reference or pointer to value in one shot.
25324	*Summary:* Make OCCT collections copy-constructible. The possibility to use copy constructor has been provided in `TCollection_Sequence` class.
25329	*Summary:* `ExprIntrp_GenExp` cannot parse unary plus. A formatting issue has been fixed in `ExprIntrp_GenExp`.
25348	*Summary:* Method `Assign` of `NCollection` containers must not change own allocator of the target. The implementation of methods `Assign` in `NCollection` package has been revised to avoid replacing the allocator of the target list with the allocator of the source list, which may cause application crash if the life cycle of the target list is longer than that of the source list. Additionally, all `NCollections` have been protected against assignment to itself with operator `=()`.

25396	*Summary:* Crash when using TBB allocator on Intel architecture not supporting SSE2 instructions. It is now checked if SSE2 instructions are supported when `MMGT_OPT=2` is in effect during the initialization of memory manager. If not, `MMgrRaw` is used instead of `MMgrTBBalloc`. This avoids a runtime crash when running on a CPU that supports SSE but does not support SSE2 (some modifications of AMD Sempron).
25401	*Summary:* Warnings in `Standard_CString.` `Standard_CString` has been modified to avoid redefining macro `strcasecmp` if it is already defined on Windows.

Modeling Data

23886 24127 24959	*Summary:* `GCPnts_TangentialDeflection` does not implement functionality by its meaning in case of a BSpline with local splash. The method `GCPnts_TangetialDeflection::PerformCurve` has been improved to divide a curve into `C_N` intervals, on which sample points are computed.
24919	*Summary:* `ShapeFix` can crash due to improper iterator handling. The variable `inter` in `Approx_SameParameter::Build` is now checked to stay in the valid range.
24945	*Summary:* `Extrema_ExtPElC::Perform` does not consider angular tolerance when calculates angle between two vectors. The class `Extrema_ExtPElC` has been modified to take into consideration angular tolerance when an angle between two vectors is calculated. It is done to protect against deviations that are significantly less than tolerance.
25011	*Summary:* `IntAna_QuadQuadGeo` can crash with `out of bounds` exception The algorithm `IntAna_QuadQuadGeo` finding geometric intersections between two natural quadrics now checks the number of intersections
25105	*Summary:* Remove `TColgp_DataMapOfIntegerCirc2d.` Obsolete class `TColgp_DataMapOfIntegerCirc2d` has been removed.
25116	*Summary:* `BRepTools` – do not force `Message_ProgressIndicator` updates at each minor iteration. The update of `Message_ProgressIndicator` is now skipped at minor iterations, where it does not have much sense.
25439	*Summary:* Enable shape binary persistence without OCAF. `BinTools` have been moved from toolkit `Bin` to toolkit `BRep` to enable shape binary persistence without OCAF.

Modeling Algorithms

23511	*Summary:* Function `BRepTools::UVBounds` provides incorrect result for a face. It has become possible to compute UV-bounds for faces based on analytic surfaces such as cone, sphere, cylinder etc. with more precision. The following modifications have been implemented to this end: ▪ UV-Bounds for outbound faces can be computed incorrectly, i.e. the first parameter can be greater than the last one. To avoid it, control of computed bounds has been extended in method `BRepTools::AddUVBounds`; ▪ The segment parameters for a B-Spline curve have been adjusted in the real parametric range of the curve in method `BndLib_Add2dCurve::Add`; ▪ The algorithm of intersection between a Cylinder and a Sphere has been improved to handle the case when the sphere apex belongs to a cylinder in functions `BRepOffsetAPI_MiddlePath::Build()` and `IntTools_FaceFace::ApproxWithPCurves`. ▪ `BrepLib::SameParameter` is now called in `HLRAppli_ReflectLines` to avoid creation of bad shapes after the algorithm works.
23753	*Summary:* Test case works too long on Linux platform. Method `IntTools_ShrunkRange::Perform()` has been fixed to avoid creation of small section edges.
24639	*Summary:* Parallelization of `FillDS` part of BO. The algorithms computing geometrical interferences between arguments of Boolean Operations have been parallelized. This concerns Edge/Edge Vertex/Edge, Edge/Face, and Vertex/Face interferences. Draw command `bfillds` that tests partition commands now works in parallelized mode by default, but can be used with option `-s` in sequential mode. The parallel mode of Boolean Operations is disabled by default; this can be changed by calling method `BOPAlgo_Algo::SetParallelMode()`.
24157	*Summary:* Parallelization of assembly part of BO. The following aspects of Building (Assembly) part of Boolean Operations have been parallelized in this version: ▪ Post-processing part of the Builder in class `BOPTools_AlgoTools`. ▪ Computation of P-curves in classes `BOPAlgo_PaveFiller` and `BOPTools_AlgoTools2D`. ▪ Building of solids with many internal faces in class `BOPAlgo_BuilderSolid`.
24513	*Summary:* Suspicious code. Redundant parameters `ff2` and `anIsProj2` have been removed in method `IntTools_EdgeFace::FindProjectableRoot`. Method `ProjLib_ProjectedCurve::Load` has been modified to make the processing of a surface of revolution symmetric.

24572	*Summary:* Improve performance of `BRepExtrema_DistShapeShape`. Function `DistanceMap` has been revised to improve the performance of `BRepExtrema_DistShapeShape`.
24608 24946 25058	*Summary:* Global optimization methods for a multivariable function The following modifications have been introduced to develop global optimization methods and use them to search for minimal distances between curves: ▪ New minimization algorithm `math_GlobOptMin` has been implemented; ▪ The classes `Extrema_GlobOptFuncCC`, `Extrema_ExtCC` and `Extrema_ExtCC2d` have been modified to implement this algorithm for curve / curve extrema; ▪ `Extrema_CurveCache` class has been deleted as obsolete code; ▪ Processing of extrema `math_NewtonMinimum` has been fixed in class `ChFi3d_Builder`.
24799	*Summary:* `BRepAlgoAPI_Common` returns empty result. Method `BOPAlgo_PaveFiller::PostTreatFF` has been modified to add newly created vertices to `myShapesSD` map for modification tracking purposes.
24807	*Summary:* Exception in `ShapeAnalysis_FreeBounds::ConnectEdgesToWires`. The algorithm `ShapeAnalysis_FreeBounds` has been improved to handle cases when a compound contains only edges with `INTERNAL` orientation without any wires or faces.
24817	*Summary:* Cannot sew two circular faces in non-manifold mode. The usage of `IndRef` parameter has been fixed in function `BRepBuilderAPI_Sewing::MergedNearestEdges`.
24829	*Summary:* `sprops` with precision throws an exception. An exception has been fixed in class `BRepGProp_Sinert`. It was caused by reading a non-existent array element.
24832	*Summary:* Performance of new Boolean operations has become worse Handles have been implemented instead of `C` pointers and encapsulated in classes `Extrema_ExtCS`, `Extrema_ExtPS`, `Extrema_ExtPExtS` and `Extrema_ExtPRevS`.
24876	*Summary:* Protect shapes against modification of geometry. New flag *Locked* has been implemented in classes `TopoDS_TShape` and `TopoDS_Shape` for protection of geometric data in the shapes (including tolerances). This flag is used by `BRep_Builder` class.
24879	*Summary:* Wrong result of General Fuse operation. New static function `MinStep3D` has been implemented in method `BOPTools_AlgoTools::GetFaceOff` for the step in 3D that is used to define angles between faces.

24880	*Summary:* Invalid result of pipe creation. Sweeping of shells or compounds of faces has been corrected in classes `Sweep`, `PipeShell`, `Draft` and `Pipe` from `BRepFill` package.
24886	*Summary:* `BRepOffsetAPI_NormalProjection` failure. Some checks have been added in the algorithm `ProjLib_CompProjectedCurve` searching for the normal projection of a wire on shape to avoid exceptions when the projection does not exist.
24889	*Summary:* `Geom2dAPI_InterCurveCurve` produces result with parameter outside the curve limits. Variable `isOutOfRange` has been introduced in `IntCurve_IntConicConic` class. If the found intersection point is out of range, the minimal distance between ends of interval is taken. If the distance is less than tolerance, this point replaces the found intersection point. Otherwise, the found intersection point is ignored.
24896 24979	*Summary:* Optimize `Extrema_GenExtCS`. The algorithm choosing the initial approximation for Newton's method has been redesigned in class `Extrema_GenExtCS`. Now the algorithm does not search for a point on the fine shifted grid. It performs the initial search for candidate points in the original coarse grid (which is cached in new version). After that particle swarm optimization (PSO) is used to localize the global minimum. This algorithm optimizes a problem by having a population of candidate solutions ("particles"), and moving these particles around in the search-space according to a simple mathematical formula taking into account the position and the velocity of a particle. The movement of each particle is influenced by its local best known position and the best known positions in the search-space updated as far as better positions are found by other particles. This strategy has reported good results in solving complex global optimization problems and allows reducing the number of evaluations in 5-10 times.
25086	*Summary:* Implementation of PSO in package `math`. The implementation of Particle Swarm Optimization method has been generalized and moved to `math` package. The corresponding classes `math_PSOParticlesPool`, `math_PSOParticlesPool` and `math_BullardGenerator` have been added.
24899 25413	*Summary:* Too slow intersection points computation with class `BRepIntCurveSurface_Inter`. The following improvements have been introduced to reduce the computation time of intersection points: ▪ Sorting by intersection of bounding boxes of faces with line or box built for curve was added in class `BRepIntCurveSurface_Inter`. ▪ New method `BRepIntCurveSurface_Inter::Init` now initializes the algorithm by a separate curve for intersection of shape by multiple curves. ▪ The possibility to intersect shape by multiple curves has been added in DRAW command `brepintcs`.

24910	*Summary:* Offset of solid is not created. Processing of sharp corners with tangent edges has been corrected in class `BRepOffset_Offset`. Building of spherical faces corresponding to sharp corners of the initial shape has been corrected in class `GeomFill_CircularBlendFunc`.
24913	*Summary:* Unused declaration. Unused local variable has been removed from method `BRepOffsetAPI_ThruSections::CreateSmoothed()`.
24914	*Summary:* Micro edge is created during Boolean Operations. The method `IntAna_QuadQuadGeo::Perform` now works correctly when the value of angle between planes is small and the origin of intersection line should be refined. The method `IntTools_FaceFace::Perform` now correctly processes cases when both faces are based on planar surfaces. The UV-bounds have been extended to provide correct solutions as it has been done for all other cases with plane-based faces.
24915 25292	*Summary:* Wrong intersection curves between two cylinders. New algorithm computing Cylinder-Cylinder intersection has been implemented in classes `IntPatch_Intersection` and `IntPatch_ImpImpIntersection`. Its main advantage consists in that it uses an adaptively computed step to obtain the Walking-line and builds it more carefully compared to the previous algorithm as the points, where the Walking-line should be split, are determined more precisely. However, this method works badly for non-trimmed cylinders (with infinite bounds), because the step value depends on the values of boundaries. Moreover, this algorithm suits only for non-analytical cases (i.e. the intersection line is not a circle, ellipse, line etc.). In these cases previous method is still used.
24920	*Summary:* Invalid result of Pipe construction. The check if a trimmed curve is closed has been implemented in method `BrepFill_ShapeLaw::Init`.
24921	*Summary:* `ShapeAnalysis_Curve::ValidateRange` does not adjust the range for periodic curves. The sequence of conditional checks has been fixed in method `ShapeAnalysis_Curve::ValidateRange`. Now the algorithm always tries to adjust the range for periodic curves.
24922	*Summary:* `ShapeAnalysis_Wire::CheckIntersectingEdges` does not report an intersection point if it is inside of tolerance even only on edge. The check of tolerances for an intersection point has been fixed in method `ShapeAnalysis_Wire::CheckIntersectingEdges`.

24924	*Summary:* `ShapeFix_SplitTool` does not verify the new range after cutting an edge. Method `ShapeFix_SplitTool::CutEdge` has been modified to validate a new cut range before setting it to the edge.
24933	*Summary:* Inconsistent results of self-interference checker in SALOME and DRAW. The methods `BOPAlgo_ArgumentAnalyzer::TestSelfInterferences()` and `BOPAlgo_CheckerSI::Perform()` have been modified to make the results of SALOME Application and DRAW Application mutually consistent. New options have been added for `bopcheck` command: ▪ `-t` displays elapsed CPU time ▪ `-s` runs the command in serial (non-parallel) mode. The default mode is parallel.
24934	*Summary:* Section of two faces causes a crash. Method `ShapeAnalysis_Wire::CheckSmall` has been protected against null vertexes.
24939	*Summary:* Incorrect result of Fuse operation. Two new static functions `IntTools_EdgeEdge::Resolution` and `IntTools_EdgeEdge::ResolutionCoeff` have been implemented to correctly compute the resolution of Hyperbola and Parabola curves.
24940	*Summary:* Cyclic dependency detected between `BOPInt` and `IntTools.` The following modifications have been done to avoid cyclic dependency between `BOPInt` and `IntTools`: ▪ The package `BOPInt` has been removed. ▪ New classes `IntTools_Context` and `IntTools_ShrunkRange` have been implemented to replace `BOPInt_Context`, `BOPInt_ShrunkRange` and `BOPInt_Tools` classes. ▪ Classes from `BOPAlgo`, `BOPTools`, `BrepFill` and `IntTools` packages have been modified to use the new classes.
24949	*Summary:* Segmentation Violation during pipe creation. New output value `ErrorOnSurface(me)` has been implemented in classes `BRepFill_PipeShell`, `BRepOffsetAPI_MakePipe` and `BRepFill_PipeShell` to improve the diagnostics of incorrect result obtained after pipe creation.
24950	*Summary:* Wrong result of Boolean CUT operation. The method `BOPAlgo_PaveFiller::PutBoundPaveOnCurve` has been modified to create bounding paves for each section curve separately.

24952	*Summary:* Possibility to break the execution of Boolean operations. The possibility to stop the execution of BOP algorithm at the user's request has been implemented using methods `BOPAlgo_Algo::UserBreak()`, which breaks the execution if the break signal is indicated, and `BOPAlgo_Algo::SetProgressIndicator`, which sets the Progress Indicator object. Support of break signal has been enabled in `BOPAlgo_Builder`, `BOPAlgo_BOP`, `BOPAlgo_BuilderFace`, `BOPAlgo_BuilderSolid` and `BOPAlgo_PaveFiller`.
24953	*Summary:* Draw crashes. The function `GCPnts_QuasiUniformAbscissa::Initialize`, which divides a given curve into several parts with equal length and returns an array of parameters in the control points, has been fixed.
24964	*Summary:* `ThruSections` crashes DRAW.exe The tolerance for concatenation of edge curves is now calculated more accurately in class `BRepOffsetAPI_ThruSections`.
24973	*Summary:* Incorrect PCurve construction. The method `BOPTools_AlgoTools2D::MakePCurveOnFace` has been modified to provide conformity with location of the surface.
24981	*Summary:* `IntTools_FaceFace` enters into infinite loop. New function `IntTools_Tools::AdjustPeriodic` has been implemented for fast adjustment of pcurves to the range of surface.
24985	*Summary:* Control the maximum degree and number of segments in Pipe Shell algorithm. It has become possible to define the maximum V degree of the resulting surface and the maximum number of spans in V-direction on the resulting surface for the Pipe Shell algorithm using methods `SetMaxDegree` and `SetMaxSegments` from class `BRepOffsetAPI_MakePipeShell`.
25002	*Summary:* Wrong result produced by Boolean Operation algorithm. The processing of lines with only one point before or after the boundary point has been improved in class `IntTools_FaceFace::DecompositionOfWLine`.
25004	*Summary:* Incorrect curve/curve extrema. The classes `Extrema_GenExtCC` and `Extrema_ExtCC` have been modified to improve extrema clustering algorithm. New function `SetTol` from `math_GlobOptMin` class has been implemented to allow changing tolerances.
25014	*Summary:* `ShapeAnalysis_WireOrder` produces `Standard_RangeError` on an empty wire (in debug mode only) An exception has been fixed in method `ShapeAnalysis_WireOrder::Perform`.

25021	*Summary:* New option of `BRepOffsetAPI_MakeOffset` algorithm: open result for an open wire. New option `IsOpenResult` has been implemented in the algorithms of offset wire construction to allow building an open offset wire for an open initial wire. By default, the resulting offset wire is closed for both open and closed wires.
25028	*Summary:* `BRepAlgo::ConcatenateWire` raises an exception. Processing of wires consisting of only one edge has been corrected in class BRepAlgo::ConcatenateWire.
25043	*Summary:* Find which sub-shape of the source shape was detected as causing problems by `BRepAlgoAPI_Check.` Copying of `BRepAlgoAPI_Check` arguments is now avoided as it is possible to run the check on self-intersection using `BOPAlgo_CheckerSI` algorithm in a non-destructive mode without modification of the source shape.
25068	*Summary:* Wires with incorrect `Closed` flag. The method `ShapeBuild_Edge::CopyReplaceVertices` now produces wires with correctly defined properties.
25095	*Summary:* Wrong result of projection algorithm The precision value has been modified in class `ProjLib_ComputeApprox` to get better projection results.
25100	*Summary:* Self-intersection of surfaces with `GeomAPI_IntSS` rises run-time check failure Static array has been replaced by a dynamic one in class `IntPatch_InterferencePolyhedron` to avoid run-time check failure.
25111	*Summary:* Incomplete section curve between Conical and Toroidal surfaces. The method `IntPatch_ImpPrmIntersection::DecomposeResult` has been improved to produce correct section curves.
25127	*Summary:* Wrong result of General Fuse algorithm. Method `BOPTools_AlgoTools::IsSplitToReverse` has been modified to reverse normal direction for REVERSED faces.
25128	*Summary:* Memory leak in `BOPDS_DS::Paves()`. The class `Ncollection_Array1` has been used to provide proper allocation and de-allocation of memory in methods `BOPDS_PaveBlock::Update` and `BOPDS_DS::Paves`.

25163	*Summary:* Wrong result of common operation for faces. The contents of a 2D Domain for the straight line have been changed in function `BOPAlgo_WireSplitter::RefineAngle2D`. The processing of cylindrical surfaces now takes into account the existing tolerance values of edges in method `BOPTools_AlgoTools2D::AdjustPCurveOnFace`.
25175	*Summary:* `BRepBuilderAPI_Sewing` can crash if an edge without 3D curve is present. The check if a 3D curve exists has been added in methods `BRepBuilderAPI_Sewing::EvaluateDistances` and `BRepBuilderAPI_Sewing::Cutting` to avoid possible Access Violation exception.
25184	*Summary:* Non-deterministic order of wires in the result of General Fuse. Basic hashed maps have been replaced by indexed maps in method `BOPAlgo_BuilderFace::PerformAreas()`.
25192	*Summary:* Unification of `LineConstructor` algorithms used by `GeomInt_IntSS` and BOP. Class `IntTools_LineConstructor` has been replaced by `GeomInt_LineConstructor` and removed. Method ::`AdjustPeriodic` has been moved from `IntTools_Tools` to `GeomInt`.
25194 25374	*Summary:* It is necessary to make transformation matrix orthogonal in `gp_Trsf` and `gp_Trsf2d` classes. The following modifications have been implemented to ensure that the transformation matrix is always orthogonal: ▪ Method `Orthogonalize` that makes the matrix orthogonal has been implemented for classes `gp_Trsf2d` and `gp_Trsf`. ▪ Method `gp_Trsf::SetValues()` now does not accept precision parameters and always makes the matrix orthogonal (instead of raising exception if it is not orthogonal with specified precision). ▪ New method `gp_Trsf2d::SetValues()` defines the coefficients of the transformation. ▪ The output of `AppParCurves_MultiPoint::Dump` method has been corrected.
25084	*Summary:* Incorrect `PCurve` construction. The choice of projected point has been improved in `ProjLib_BuildInitialCurve2d`.
25109	*Summary:* Check `PolygonOnTriangulation` contained in edges. New method `BRepCheckEdge::CheckPolygonOnTriangulation` has been added to check if the polygon on triangulation of `theEdge` is out of 3D-curve of this edge. New status `BRepCheck_InvalidPolygonOnTriangulation` is assigned to faulty polygons.

25174	*Summary:* Provide `ShapeCustom` and `BRepModifier` with Progress Indicator and Reshape features It has become possible to track the progress and changes done by `ShapeCustom` and `BRepModifier` algorithms.
25199	*Summary:* Bad tolerance of an edge generated by `blend` algorithm. The method `GeomInt_IntSS::DecompositionOfWLine` has been corrected to fix surface borders computation.
25202 25365	*Summary:* Incorrect value of `IsClosed` flag in shapes produced by some algorithms. The method `BRep_Tool::IsClosed()` has been extended to analyze closure of wires in addition to shells and solids. This check ignores external and internal edges and vertices and does not analyze compounds. The flag `Closed` is now updated whenever a new shell is constructed.
25207	*Summary:* `GCPnts_QuasiUniformDeflection` achieves incorrect result on some curve. A misprint in method `GCPnts_QuasiUniformDeflection::PerformCurve` has been fixed.
25210	*Summary:* Wrong result of conical projection. The algorithm of conical projection of a wire onto a face has been corrected in `BrepProj_Projection` class.
25223	*Summary:* Wrong result of projection algorithm. The class `ProjLib_ComputeApprox` now uses `Precision::PConfusion()` as 2D tolerance when shifting the projected curve.
25224	*Summary:* Section curve between two cylindrical faces is incomplete The algorithm seeking point on boundaries has been amended in method `IntPatch_ImpImpIntersection::IntCyCyTrim`.
25225	*Summary:* Failure of 2d offset algorithm on two wires with arc. Precision is now taken into account in method `BrepFill_OffsetWire::MakeOffset.`
25228	*Summary:* Wrong result of General Fuse operation for an edge and a face. Processing of seam edges for periodic surfaces has been improved in method `BOPTools_AlgoTools3D::DoSplitSEAMOnFace`.

25232	*Summary:* Functionality to create solids from a set of shapes. The class `BOPAlgo_MakerVolume` has been implemented to build solids from set of shapes. It uses `BOPAlgo_Builder` algorithm to intersect the given shapes and build the face images and `BOPAlgo_BuilderSolid` algorithm to build solids.
25237	*Summary:* Wrong result of `COMMON` operation. The algorithm checking closed edges for intersection has been corrected in class `IntTools_EdgeEdge`.
25242	*Summary:* Wrong result of cut operation. The following modifications have been introduced to improve Cut operation: ▪ Methods `Perform*Z() of the class BOPAlgo_CheckerSI` have been moved to class `BOPAlgo_PaveFiller`. ▪ Processing of interferences with solids has been added in method `BOPDS_Iterator::Intersect();` ▪ Empty interferences are now not added in the table of interferences In method `BOPAlgo_PaveFiller::PerformFF()`.
25243	*Summary:* `BrepFeat_SplitShape` algorithm processes incorrect faces with degenerated edges. Class `LocOpe_ProjectedWires` has been replaced by `LocOpe_WiresOnShape` to improve the shape splitting.
25245	*Summary:* General Fuse operation regression. The processing of internal edges has been corrected in method BOPDS_DS::InitPaveBlocks()
25248	*Summary:* Curve-Surface intersection algorithm raises an exception. Curve-Surface intersection algorithm has been improved in method `Intf_Tool::Inters3d`.
25254	*Summary:* Protection on `tgtfaces.` The check if pcurves exist on both faces of the edge has been implemented in function `tgtfaces` from `BrepLib` class.
25256	*Summary:* Small optimization in the algorithm converting Bezier curves to Bspline. The method `CompBezierCurvesToBSplineCurve::Perform` has been optimized.
25258	*Summary:* Uninitialized class field in `IntPatch_CSFunction.` The method `IntPatch_CSFunction::IntPatch_CSFunction` has been modified to properly initialize field `f`.
25259	*Summary:* Incorrect split on a toroid surface. The algorithm `LocOpe_SplitShape` has been improved to correctly process cases when several alternative directions of wire creation are possible.

25263	*Summary:* Wrong result of cut operation. Processing of internal edges has been enabled in method `BOPDS_DS::ChangePaveBlocks`.
25270 25407	*Summary:* Exception in extrema operation. The method `Extrema_GenExtCS::Perform` has been fixed to correctly determine the starting point of extrusion in case of infinite border curves.
25272	*Summary:* Invalid shape is created by `BRepOffsetAPI_MakePipe`. The check for edge orientation has been introduced in `BrepFill_Sweep.`
25285	*Summary:* Wrong result of General Fuse operation for an edge and a face. `BOPTools_AlgoTools2D::AdjustPCurveOnFace` the adjustment for 2D curve now uses precision value in parametric space.
25298	*Summary:* New option of `BRepOffsetAPI_MakeOffset` algorithm: processing of sharp corners in mode `GeomAbs_Intersection,` New parameter `JoinType` has been implemented in method `MAT2d_Circuit::IsSharpCorner`. ▪ If this parameter is equal to `GeomAbs_Intersection` the sharp corners are offset by prolongation until intersection of adjacent edges; ▪ If it is `GeomAbs_Arc,` the vertices generate sections of a circle. `JoinType` parameter has been also added in Draw commands `mkoffset` and `openoffset`.
25319	*Summary:* BOP Common produces strange results with same shapes. The method `BOPAlgo_Builder::FillIn3DParts` has been improved to sort faces before classification relatively to the solid.
25334	*Summary:* `BRepOffsetAPI_MakeOffset` algorithm crashes. Processing of open wires with option `OpenResult` has been corrected in method `BRepMAT2d_BisectingLocus::Compute`.
24337 25420	*Summary:* Boolean operations fail on two planar circular faces lying in the same plane. Precision of the algorithm looking for intersection between edges has been increased in method `IntTools_EdgeEdge::FindSolutions`.
25352	*Summary:* Improve precision of analytical calculation of fillets 2d:. The algorithm `ChFi2d_AnaFilletAlgo,` which calculates the fillet arc between two edges and modifies the edges shrinking them by the fillet, has been modified to reuse the points already calculated by the fillet algorithm to shrink the neighbors by these points. This improves the tolerance of calculations (by reusing points shared by the edges).

25354	*Summary:* New intersection operation. New class `BOPAlgo_Section` provides an Intersection algorithm finding common part of two input objects. Unlike the Boolean operation `Common` that requires shapes of the same order as parameters and result; the new operation provides the result even if the shapes of different type are specified and can produce the shape of a lower type as a result.
25355	*Summary:* Recognizing and merging the model from STEP leads to crash. The problem with empty handle usage, which caused a crash of Canonical Recognition, has been fixed in method `BRepTopAdaptor_TopolTool::Classify`.
25368	*Summary:* `BREPExtrma DistShapeShape` gives wrong result for Sphere and Line. The analytical Sphere/Line and Cylinder/Line extrema algorithms have been modified in `Extrema_ExtElCS` class to search for perpendicular and intersection points.
25376	*Summary:* Inconsistence between function and derivatives evaluation in `Extrema_GlobOptFuncCS.` The `value` function from `Extrema_GlobOptFuncCS` class has been enabled to compute square distance between a point on curve and a point on surface.
25380	*Summary:* Intersection curve cannot reach surface boundary. The algorithm of next point computation has been changed in IntWalk_PWalker. Now it tries to walk over another isoline if a new point is too close to the previous one.
25406	*Summary:* The algorithm `BRepOffset_MakeOffset` fails on a face with two degenerated edges. The method `BRepOffset_MakeOffset::CorrectConicalFaces` now can process the case when the initial shape has faces with singularities along V-direction (null u-iso curves).
25408	*Summary:* Wrong result obtained by General Fuse operator. The method `BOPTools_AlgoTools2D::AdjustPCurveOnFace` has been corrected to adjust the location of 2D-curve to face boundaries using the precision value in a parametric space.
25427	*Summary:* The algorithm building a plane from wire hangs. The algorithm building a plane from wire has been fixed in class `BRepLib_FindSurface`.
25446	*Summary:* `BRepAlgoAPI_BooleanOperation::IsDeleted()` returns `TRUE` for the faces contained in the result of BOP. The method `BOPAlgo_Builder::IsDeleted` has been modified to return TRUE only if the shape `theS` has been deleted.

Visualization

2883	*Summary:* It is impossible to set material, color and transparency to compound. The shading presentation now can be updated faster when color, material or transparency is set. The algorithm implemented for `AIS_Shape` now iterates through all groups (not only the last one) and through all presentations to change presentations in all viewers.
18942 24001 24381 24714 24717 24808 24882 25137 25301	*Summary:* Stereographic rendering support. The following features have been implemented to provide stereographic rendering support: ▪ Support for two-pass stereo-rendering at separate left and right buffers at `TKOpenGL` level. ▪ Calculation of appropriate projection and view orientation matrices for left and right eyes. ▪ Notion of 3D View Camera at high-level API, which sets up the desired projection in a convenient way.
22240	*Summary:* Bad triangulation of transformed shapes. The method `BRep_Tool::CurveOnSurface` that transforms curves on a plane surface has been corrected to improve presentation of transformed shapes in shading display mode.
23235 24521 24672	*Summary:* Automatic back face culling is not turned on for Solids packed into compound. Culling of back faces reduces the number of triangles rendered (improving the performance) and eliminates artifacts at shape boundaries. However this option might be used only for solid objects, where the interior actually should not be visible from any point of view. Automatic back-face culling mechanism is turned on by default, controlled by `V3d_View::SetBackFacingModel()`. The following issues have been fixed in `StdPrs_ToolShadedShape::IsClosed()`, which is used for definition of back face culling in Shading Aspect: ▪ disable culling for free closed Shells (not inside the Solid) since reversed orientation of free Shell is valid case ▪ enable culling for Solids packed into compound ▪ ignore Solids with incomplete triangulation Back face culling is now turned off at `TKOpenGl` level in the following cases: ▪ clipping/capping planes are in effect ▪ for translucent objects and objects with hatching presentation style

23422 24756 24837 25103 25335 25366 25370 25371 25373	*Summary*: Revise design and implementation of connected Interactive Objects. The following changes have been introduced in the API to improve the design of connected Interactive Objects: ▪ Every `PrsMgr_PresentableObject` may have its child objects which inherit transformation. ▪ Combined transform (with parent object) of `PrsMgr_PresentableObject` may be accessed with `Transformation()` method. ▪ Children objects of `PrsMgr_PresentableObject` may be accessed as NCollection_List of presentable objects with `Children()` method. `AddChild()` and RemoveChild() methods can modify this list. ▪ `AIS_ConnectedInteractive` is now an instance of object. It reuses geometry of connected object but has its own transformation, material, visibility flag etc. This connection is propagated down to `OpenGl_Structure`. ▪ `Connect()` methods from `AIS_ConnectedInteractive` and `AIS_MultiplyConnectedInteractive` now use `gp_Trsf` object instead of `TopLoc_Location`. ▪ Behavior of `AIS_ConnectedInteractive.Connect()` is almost the same as before despite the fact that a newly created instance will become the child of original object in scene hierarchy and will inherit the transformation. It will still be connected after `RemoveChild()` call. ▪ `AIS_ConnectedInteractive` can be connected to any `AIS_Interactive` object in general. When it is connected to another `AIS_ConnectedInteractive`, it copies a reference to the original object. ▪ `AIS_MultiplyConnectedInteractive` represents an assembly, which doesn't have its own presentation. Assemblies are able to participate in scene hierarchy and are intended to handle a grouped set of instanced objects. An assembly behaves as a single object in terms of selection. ▪ `AIS_MultiplyConnectedInteractive` stores instances (`AIS_ConnectedInteractive`) to its connected objects in `Children()` list. It applies high level transform to all sub-elements since it is located above in the hierarchy. ▪ `AIS_MultiplyConnectedInteractive` never moves, deletes or copies objects with its `Connect`() method. It only creates instances. ▪ When one `AIS_MultiplyConnectedInteractive` is connected to another, a new `AIS_MultiplyConnectedInteractive` will be created and all instances will be copied. ▪ New method `AIS_MultipleConnectedInteractive::Connect()` takes transformation persistence flags, i.e. they are copied from original objects when `Connect()` is called without extra arguments.
23649	*Summary:* `AIS_LocalContext` - make highlighting of already selected objects consistent with and without Shift modifier. New methods `SelectMgr_EntityOwner::IsSelected()` and `SelectMgr_EntityOwner::SetSelected()` allow determining if the corresponding entity is selected. They replace the functionality of `SelectMgr_EntityOwner::State()` methods that are now deprecated. New methods `AIS_InteractiveContext::ToHilightSelected()` and `AIS_InteractiveContext::SetToHilightSelected()` can be used to enable highlighting of selected objects. By default the selected objects are highlighted, which is a new behavior. New Draw command `vhighlightselected` has been added to enable/disable 'highlight selected objects' mode.

23710	*Summary:* Simplification of presentation management classes. The class `PrsMgr_Presentation3D` has been merged into `PrsMgr_Presentation`.
23804	*Summary:* Unexpected color modulation during texture mapping. `AIS_TexturedShape` has been modified to create `Graphic3d_AspectFillArea3d` instance with the material defined by the Drawer, instead of the default material.
23814	*Summary:* Drop plugin interface for `Graphic3d_GraphicDriver` instantiation. Instantiation of `OpenGl_GraphicDriver` using a dynamically loaded library has been eliminated. The following elements have been removed: ▪ Methods `Graphic3d_GraphicDriver::Begin()` and `::End()` (always performed right after driver instantiation and before destruction). ▪ Dummy argument for `Graphic3d_GraphicDriver` constructor with library name. ▪ `Graphic3d::InitGraphicDriver()` function. The application now explicitly links against `TKOpenGl` toolkit and instantiates `OpenGl_GraphicDriver` class. ▪ `MetaGraphicDriverFactory` implementation from `TKOpenGl`. ▪ `CSF_GraphicShr` from the generated `env.bat` in WOK.
24287 24413 24606 24711 24720	*Summary:* Visualization avoid projection shift in orthographic camera definition. The projection shift concept was removed. The reference system of the view is now located at the screen center, in consistence with definition of a camera. See the details in New Features section. The following extra modifications have been introduced: In `V3d_View` class: ▪ `V3d_View::Center()` has been removed. The method `SetCenter()` now changes relative location of the screen center of. ▪ Panning now translates `At, Eye` position of the view along the view plane. ▪ `::SetViewingVolume()` has been removed. The same modifications to viewing volume can be done via camera of the view `Camera()`. ▪ Duplicating method `Zoom (CurrX, CurrY)` has been removed. The method `Zoom (FromX, FromY, ToX, ToY)` should be used instead. ▪ `SetAutoZFitMode(), AutoZFitScaleFactor(), ZfitAll()` deal with new parameter: `Z-range` margin. ▪ The method `Turn()` now performs rotation around view reference system, which is located at center of the screen (around screen axes). To produce a rotation similar to the effect of `Center()`, the view can be rotated around a point with `Rotate()` method. In `Graphic3d_Camera` class: ▪ `ProjectionShift()` and `SetProjectionShift()` have been removed. ▪ `SetZNear()` and `SetZFar()` have been replaced by unified `SetZRange()` method. `Znear` and `Zfar` can be negative for orthographic camera. ▪ `BeginUpdate()` and `EndUpdate()` methods have been replaced by matrix lazy-computation approach. There is no necessity in these methods anymore. ▪ Redundant `WindowLimit()` method has been removed. It had the same meaning as `ViewDimensions()`, summed up with `ProjectionShift()`. ▪ The type of argument in `ViewDimension()` and `AxialScale()` has been changed from `gp_Pnt` to `gp_XYZ`.

24287 24413 24606 24711 24720	In `Select3D_Projector` class: ▪ Methods have been added to define both view-transformation and projection-transformation matrices. ▪ The projector now contains constant projection / orientation state. Previously it was possible to pass handle on the view, so the `Project()` method produced projection results for the actual view projection. However, the same handle was not used for other projector methods: `Transform()` and `Transformation()`. ▪ `SetView()` captures the current state of its projection / orientation matrices instead of saving a reference on the passed view.
24307 25036 25251	*Summary:* `TKOpenGl` – efficient culling of large number of presentations. The algorithm of frustum culling on CPU-side has been implemented and activated by default for 3D viewer. This algorithm allows skipping presentation outside camera at rendering stage, providing better performance. The following modifications have been introduced made to support this method: ▪ `CALL_DEF_BOUNDS`, `CALL_DEF_BOUNDBOX` and `Graphic3d_CBounds` have been replaced by a single representation for axis-aligned bounding box, `Graphic3d_BndBox4d` / `Graphic3d_BndBox4f` (depending on vector type); ▪ `Graphic3d_Structure::CalculateBoundBox()` is now used to calculate axis-aligned bounding box of a presentation considering its transformation; ▪ `V3d_View::SetFrustumCulling` enables or disables frustum culling for the specified view; ▪ New command `vfrustumculling` enables or disables the algorithm for currently active view from Draw Test Harness; ▪ New classes `OpenGl_BVHClipPrimitiveSet` and `OpenGl_BVHTreeSelector` have been added to handle detection of outer objects and usage of acceleration structure for frustum culling. ▪ `BVH_BinnedBuilder` class splits several objects with null bounding box.
24323 24402	*Summary:* `TKOpenGl` - Implement clipping planes in Phong `GLSL` program. The number of lights has been limited in Phong GLSL program.
24354	*Summary:* `TKOpenGl` - location modification of a multi-connected shape has no effect when the object is drawn by GLSL program. The handling of locations by `GLSL` program has been corrected in class `OpenGl_ShaderStates`.
24503	*Summary:* `TKOpenGl` – Porting ray-tracing component to `BVH` package. The preparation of geometry for the ray-tracing engine is now done using two-level BVH provided `BVH` package. Two-level hierarchy provides efficient dynamic updates and instancing. Individual bottom-level BVHs are built for each ray-tracing object (`OpenGL` primitive array), then a high-level BVH is built over these sets of objects. This reduces the overall tree quality, but provides fast updates when objects are added or removed (already built BVHs can be reused). Independent bottom-level BVH can be constructed in parallel.
24520	*Summary:* Implement affine transformations in ray-tracing. The classes `OpenGl_Workspace`, `OpenGl_Workspace_Raytrace` and `OpenGl_SceneGeometry` have been modified to apply all GPU ray-tracing transformations on-the-fly, which provides ray-tracing animated scenes and scene editing in real-time.

24534	*Summary:* Improve design of `Image_PixMap` class. `Image_PixMapData` has become a non-template class. The following changes have been implemented: ▪ Template methods `Image_PixMap::EditData()` and `::ReadData()` have been removed. Template method `Image_PixMap::ChangeValue()` has been added. ▪ Redundant parameter for `Image_PixMap::Clear()` method has been removed.
24546 24739 24795 25414	*Summary:* `TKOpenGl` - port ray-tracing from `OpenCL` to GLSL for better integration and portability. OCCT ray-tracing core has been redesigned and ported to OpenGL/GLSL framework to improve stability and extend supported GPU devices. Currently, most of OpenGL 3.1 compatible cards can run ray-tracing. The rendering performance has also been improved: ray-tracing performance has increased by up to two times in "lightweight" (only shadows or transparency) modes.
24590	*Summary:* Crash when processing `OpenGl_BndBoxPrs` objects. OpenGL package has been revised to remove a wrong type cast.
24610	*Summary:* Skip erased objects while computing bounding box for Fit All operations. The method `Visual3d_View::MinMaxValues()` has been corrected to ignore hidden `Graphic3d_Structure` instances. Unused method `Graphic3d_StructureManager::MinMaxValues()` has been removed.
24637 24791 25190	*Summary:* Clean up implementation of rendering in immediate mode. Immediate rendering mode introduces combined rendering technique when the main scene is rendered in the back buffer, which avoids displaying intermediate rendering stage, but temporary presentations are drawn directly into front buffer (contrary to the standard rendering which presumes that all presentations are initially drawn in the back buffer). The implementation of immediate mode has been improved to address issues with usage of global variables, safety and usage limitations. The following main modifications have been introduced: ▪ `Visual3d_TransientManager` class has been replaced with `PrsMgr_PresentationManager`. ▪ Methods `::TransientManagerBeginDraw()`, `::TransientManagerClearDraw()` and `::TransientManagerBeginAddDraw()` have been removed from class `V3d_View`. ▪ Method `V3d_View::RedrawImmediate()` has been added to redraw only the layer of immediate presentations. ▪ New methods `::DisplayImmediateStructure()`, `::EraseImmediateStructure()` and `::RedrawImmediate()` have been added in `OpenGl_GraphicDriver` to manage rendering of immediate presentations.

24637 24791 25190	▪ The class `PrsMgr_PresentationManager3d` has been merged into `PrsMgr_PresentationManager`. ▪ New flag `theToRedrawImmediate` has been added in methods `::MoveTo()`, `::HilightNextDetected()` and `::HilightPreviousDetected()` from `AIS_InteractiveContext` and `AIS_LocalContext` to prevent view update. It allows update of customized immediate structures before redraw but after `MoveTo`. ▪ Redundant scene redraw call is now avoided in `V3d_View::ToPixMap()`. ▪ `PrsMgr_PresentationManager` now stores list of temporary immediate presentations automatically cleared within `BeginImmediateMode()` call. ▪ A presentation is not rendered during `ImmediateAdd()` call immediately but is added to the list of immediate presentations. ▪ Methods with ambiguous names from `PrsMgr_PresentationManager` class have been renamed: `Add` into `ImmediateAdd`; `Remove` into `ImmediateRemove`; `BeginDraw` into `BeginImmediateDraw`; `EndDraw` into `EndImmediateDraw`; ▪ `Visual3d_View` now stores the map of displayed immediate presentations.
24687	*Summary:* `TKOpenGl` shader programs – same view state for different v3d views. `OpenGl_ShaderManager` now invalidates view state depending on view change.
24704	*Summary:* Inherit `OpenGl_Structure` from `Graphic3d_CStructure.` `OpenGl_Structure` has become inherited from Graphic3d_CStructure to facilitate sharing of data on `TKV3d/TKOpenGl` layers, such as structure bounding box, transformation matrix or list of clipping planes. The following modifications have been introduced in frame of this improvement: ▪ `Graphic3d_Structure` now stores `Graphic3d_CStructure` as handle. ▪ Unused method `Graphic3d_Structure::SetManager()` has been removed. ▪ Structure-related methods have been moved from `Graphic3d_GraphicDriver` to `Graphic3d_CStructure` interface. ▪ Unsupported value `Aspect_TOHM_BLINK` has been removed. ▪ Duplicate field `OpenGl_Structure::myClipPlane` has been removed.
24723	*Summary:* Methods not implemented in `Visual3d` and `V3d.` Unused enumeration `V3d_TypeOfProjectionModel` as well as methods `V3d_View::SetProjModel()` and `V3d_View::ProjModel()` have been removed
24728	*Summary:* Tests crash on `vdump` command on Windows in debug mode. Method `V3d_View::ToPixMap` has been corrected to avoid crash on `vdump` command.
24732	*Summary:* OpenGl_Context - retrieve functions up to GL4.4. The list of provided OpenGL functions has been extended till GL4.4. The following modifications have been introduced: ▪ Unused structure `OpenGl_ArbVBO` has been removed. ▪ Structure `OpenGl_ExtFBO` has been replaced with `OpenGl_ArbFBO`. ▪ Unused fields `core12`, `core13` and `core14` have been dropped from `OpenGl_Context`; ▪ `OpenGl_ArbIns`, `OpenGl_ArbTBO` and `OpenGl_TextureBufferArb` are now also available within GL3.1+ (even if ARB extensions are unavailable).

24752	*Summary:* Inherit `OpenGl_Group` from `Graphic3d_Group`. Graphic3d_Group has become an abstract class and should be instantiated using `Graphic3d_Structure::NewGroup()` method. The following modifications have been implemented in connection with this improvement: ▪ Class `Graphic3d_Cgroup` has been removed; ▪ The groups list is no more duplicated within `Graphic3d_Structure` and `OpenGl_Structure`; ▪ Redundant field `myCurrentGroup` has been removed from `Prs3d_Presentation` class; ▪ Possible `NULL` reference is avoided in methods `PrsMgr_Presentation3d::Erase()` and `PrsMgr_Presentation3d::Clear()`; ▪ Group management methods have been moved from `OpenGl_GraphicDriver` and `Graphic3d_GraphicDriver` to `Graphic3d_Cstructure` and `Graphic3d_Group` interfaces.
24777	*Summary:* `AIS_InteractiveContext::MoveTo()` doesn't keep the detected object at the first call. The bug, which caused `AIS_InteractiveContext` to report an empty list of detected objects on the first `MoveTo()` call at a specified cursor location, has been fixed. In Draw Harness 3D viewer rectangular selection for an empty rectangle is now ignored in methods `Select` and `ShiftSelect` from `ViewerTest_EventManager`.
24797	*Summary:* `TKOpenGl, OpenGl_PointSprite:` assert is absent during incorrect object releasing. A warning about possible GPU leakage has been added in method `OpenGl_PointSprite::Release`.
24819	*Summary:* `TKOpenGl` – extend the ray-tracing core by visualization of lines, text and point sprites. The ray-tracing core has been extended to obtain the resulting image as a combination of ray-tracing output image and usual rendering for non-ray-traceable elements: ▪ Depth-less 2D Underlayer, Depth-less 2D Overlayer and Depth-less Immediate mode (currently used only within Local Selection); ▪ Non-ray-traceable 3D presentations, including texture-based text, hollow triangulation, lines and point sprites. The following modifications have been implemented: ▪ New function `OpenGl_SceneGeometry::IsRaytracedElement` checks if the element contains a ray-traceable geometry; ▪ New class `OpenGl_RaytraceFilter` has been added to filter all ray-traceable structures; ▪ `OpenGl_Workspace::Redraw()` has been extended for ray-tracing mode: all non-ray-tracable objects are rendered with help of OpenGL rasterization to the FBO, which is the input for the subsequent ray-tracing algorithm. ▪ `OpenGl_Workspace::Raytrace()` and ray-trace shaders have been improved to extend visualization of the ray-traced (shaded) elements using the frame buffer and depth buffer data prepared with help of OpenGL rasterization at the previous step. ▪ `Visual3d_View::Redraw()` automatically redraws the presentation for the second time if the device is lost.

24835	*Summary:* Graphic structure for highlighting is not erased if presentable object for highlight is deleted. The checks of highlighting state have been improved in class `PrsMgr_Presentation`.
24864	*Summary:* Implementing refractions in ray-tracing. Optical refractions according to the specified refraction index have been implemented for transparent objects in OCCT ray-tracing core. Transparent materials 'water', 'glass' and 'diamond' have been modified to look better with refractions.
24872	*Summary:* Support of emission color in Phong shader and ray-tracing. Implementation of shaders in `RaytraceBase.fs` and `PhongShading.fs` has been corrected to take into account material emission.
24855	*Summary:* Revision of parameters of standard materials The consistency and visual appearance of material parameters have been improved in various rendering modes (including ray-tracing). ▪ The specular exponents (shininess) have been significantly increased for metallic surfaces (brass, bronze, copper, gold, pewter, silver, steel, chrome and aluminum) ▪ Specular colors have been reviewed for some metals (copper, gold, aluminum and silver). ▪ Diffuse reflection of some metals (silver, aluminum and chrome) has been increased to make them brighter. ▪ Materials now have a new parameter – refraction index. Advanced rendering methods based on shaders or ray-tracing can use it to produce refraction effects. New transparent materials: 'Water', 'Glass', and 'Diamond' have been added. 'Charcoal' material has been implemented for modeling dark diffuse surfaces.
24887	*Summary:* Revise and extend Ray tracing controls The API controlling rendering options has been revised and extended (including features specific for ray-tracing). Now all rendering options are represented by `Graphic3d_RenderingParams` structure, which is available from `V3d_View` object. This structure allows switching between rasterization and ray-tracing mode and enabling/disabling such effects as shadows, reflections and antialiasing. Two new options have been provided in DRAW using new `vrenderparams` command: ray-tracing depth (controls the number of ray bounces) and transparent shadows (computes light propagation according to object transparency).
24894	*Summary:* `TKOpenGl` - refactor `OpenGl_FrameBuffer` and `OpenGl_Texture` classes. The class `OpenGl_FrameBuffer` has been extended by depth-stencil texture. `RenderBuffer` objects have been removed and two textures from `OpenGl_Texture` (color and depth-stencil) have been bound to a common frame buffer object.

24902 24903 25132	*Summary:* `StdPrs_ShadedShape` – advanced processing of Compounds containing mixture of closed Solids and open Shells. The flag `theToExploreSolids` has been added to method `StdPrs_ShadedShape::Add()`. Previously the tool collected triangulations from all Faces into a single graphic group. The automated algorithm for back-face culling activation has to disable culling when a single Compound also contains open Shells within closed Solids. Now the tool is able split these two categories of primitives into dedicated groups with independent back-face culling settings. In addition, the closed flag is now stored in `Graphic3d_Group::myIsClosed`, which permits the capping algorithm `OpenGl_CappingAlgo` to automatically filter improper groups.
24904	*Summary:* Integration of `VIS` component. New toolkit `TKIVtk` including packages `IVtkVTK`, `IVtkTools`, `IVtkOCC` and `IVtk` has been implemented to provide OCCT interface for VTK library functionality: it allows using VTK window and event management for OCCT shapes.
24926	*Summary:* `MeshVS` – improve generation of primitive arrays. The following improvements have been introduced in `MeshVS` to improve generation of primitive arrays: ▪ Mesh faces are now decomposed into triangles at `MeshVS` level. ▪ Arrays of triangles are used instead of arrays of polygons with arbitrary vertex number for shaded mesh representation, and arrays of lines are used instead of arrays of polylines in wireframe mode. Consequently, the whole mesh can be drawn with 1-2 OpenGL calls (as it is done in `AIS_Shape`). The array of lines avoids duplication of the same edges. Hash map with pair of integer keys and symmetric hasher is used for this purpose. ▪ `glDrawElements()` is now used rather than `glDrawArrays()` to avoid duplication of mesh vertex coordinates data.
24928	*Summary:* FSAA is not working in ray-tracing mode. The method `OpenGl_Workspace::RunRaytraceShaders` has been improved to enable ray-tracing adaptive full-scene anti-aliasing (FSAA).
24930	*Summary:* Set color and material for Textured Shapes. The methods `AIS_TexturedShape::SetMaterial`, `UnsetMaterial` and `UnsetColor` have been implemented to define material and color properties for textured shapes.
24965	*Summary:* Problem in local selection mode with selected objects staying in the viewer. New method `AIS_LocalContext::ClearOutdatedSelection` has been implemented to clear outdated selection and detection of owners for the interactive object.

24966	*Summary:* Problem in local selection mode with selected objects staying in the viewer after erase. The method `AIS_LocalContext::Erase` has been improved to take into account entity owners built from decomposition of the parent interactive shape when a shape is erased in the local selection context.
24996	*Summary:* Newly displayed objects are clipped until the first camera movement. The following improvements have been implemented to avoid clipping newly displayed objects if they do not fit in the bounding box of the previous scene: ▪ `AutoZFit` operation now can be applied on `Visual3d_View` level. ▪ `Visual3d_View` tracks `Graphic3d_Structure` updates and calls `AutoZFit` within `Visual3d_View::Redraw` if necessary. ▪ The method `ZfitAll` has been moved from `V3d_View` to `Graphic3d_Camera` in order to get `AutoZFit` functionality on `Visual3d_View` level. The method `AutoZFit` and flag `AutoZFitMode` are now a part of `Visual3d_View`.
25003	*Summary:* Selection of a face fails if sensitive triangulation is computed with interior flag = `false`. The length check has been added in method `Select3D_SensitiveTriangulation::S3D_STriangul_NearSegment` to avoid manipulation with a vector of null length.
25008	*Summary:* `Image_AlienPixMap` - wrong PPM generated by `::savePPM()` when image width is not equal to height. The method `Image_AlienPixMap::savePPM` has been corrected to generate PPM properly.
25017	*Summary:* Materials in Ray-tracing are messed up. Incorrect ray-tracing behavior (some structures could be hidden, and materials messed up) for connected OpenGL structures has been fixed in various classes from `OpenGl` package.
25027	*Summary:* 2D layer viewport is updated incorrectly on resize The method `Visual3d_ViewManager::Redraw()` has been improved to correctly update 2D layer viewport on resize.
25034	*Summary:* Highlighted dimension objects are not in `Detected` list of `AIS_InteractiveContext`. `AIS_InteractiveContext` and `AIS_LocalContext` classes have been fixed to make it possible to fill them with interactive objects, not with `AIS_Shape` only.
25040	*Summary:* Capping plane is drawn at wrong position when created by copy. -1 is now used instead of 0 as default uninitialized modification counter in method `OpenGl_CappingPlaneResource::UpdateTransform()`.

25047	*Summary:* Public methods of `OpenGl_AspectFace`, `OpenGl_AspectLine` and `OpenGl_AspectMarker` classes are not exported. Missing export qualifier (Windows) has been added to public methods of classes `OpenGl_AspectMarker`, `OpenGl_AspectFace`, `OpenGl_AspectLine`, `OpenGl_AspectText` and `OpenGl_ShaderManager`.
25052	*Summary:* Activation of all Clipping Planes within driver limit leads to broken planes management. The viewer has been modified to handle correctly the case when the number of defined clipping planes reaches driver limit (6 on old and 8 on recent hardware). The bug in `Aspect_GenId::Available()` reporting a wrong value has been fixed.
25060	*Summary:* Capping breaks highlighting of shaded `AIS_Shape` with displayed edges. The method `OpenGl_CappingAlgo::RenderCapping` has been modified to remember and restore depth function mode.
25063	*Summary:* 2D objects are not displayed at some camera positions. Minimum thresholds have been introduced in methods `V3d_View::ZFitAll` and `Graphic3d_Camera::SetDistance()` to avoid FPE exception caused by extremely thin camera frustum on 2D objects.
25066	*Summary:* Fit all does not work when the bounding box is set through `Graphic3d_Group::SetMinMaxValues()`. Method `Graphic3d_Group::IsEmpty()` has been improved to properly mark the bounding box as valid.
25071 23984	*Summary:* Inconsistent deflection values used by `AIS_Shape`. The function `Prs3d::GetDeflection()` now should be used in all TKV3d packages to calculate absolute deflection for a `TopoDS_Shape`. `AIS_Shape::GetDeflection()` has been removed.
25093	*Summary:* Deletion of `SelectMgr_SelectableObject` does not invalidate owners. The method `SelectMgr_Selection::Destroy` has been implemented to nullify handles to `Entity_owner` in `SelectMgr_Selection` destructor.
25098	*Summary:* `Select3D_SensitiveCurve` - fix computation of the depth. New method `Select3D_SensitiveCurve::ComputeDepth` has been implemented to compute the depth by intersection of a curve segment and the eye-line. New option `-entities` has been added to command `vstate` to display information about detected entities.

25099	*Summary:* Option to show vertices of a shape. It has become possible to draw all vertices of a shape by `AIS_Shape`. The corresponding option `VertexDrawMode` has been added to `Prs3d_Drawer` and `AIS_Drawer` classes. This option can be used to set the default vertex draw mode (through `DefaultDrawer()` of `AIS_InteractiveContext` instance) and to specify the mode individually for interactive objects. Vertex draw mode works regardless of the current display mode. Additionally, a special `Prs3d_VDM_Inherited` value is allowed at `AIS_Drawer` level. It means that the corresponding `AIS_InteractiveObject` instance inherits the value of vertex draw mode from the default drawer. To check this, `IsOwnVertexDrawMode()` method should be used.
25121	*Summary:* `OpenGl_Text` - correct scaling factor for fixed-size text at 3D point. The scaling factor for a fixed-size text at 3D point has been corrected in method `OpenGl_Text::Render` to eliminate blurring effect.
25129	*Summary:* Add interactive object for Points Cloud objects. New class `AIS_PointCloud` has been implemented for efficient drawing of large arbitrary sets of colored points. It uses `Graphic3d_ArrayOfPoints` to pass point data into `OpenGl` graphic driver for drawing them as an array of "point sprites". The point data is packed into vertex buffer object for performance. The type of point marker used to draw points can be specified as presentation aspect. The presentation provides selection by a bounding box of the visualized set of points. It supports two display / highlighting modes: points or bounding box. Additional changes: ▪ The classes `Graphic3d_ArrayOfPoints`, `OpenGl_PrimitiveArray` and `OpenGl_VertexBuffer` have been updated to use normal for points; ▪ Draw command `vpointcloud` has been added to generate a point cloud presentation for nodes of a shape's triangulation or presentation for an arbitrary set of points of given size.
25133	*Summary:* `TKOpenGl` - Crash on closing a view containing presentations with capping. `OpenGl` package has been revised to make element destruction and resource de-allocation methods accept a pointer to `OpenGl_Context` instead of a handle to avoid double deletion of the context.
25142 25307	*Summary:* Visualization breaks triangulation on shared solids in a composite solid model. `AIS_TexturedShape` has been modified to process the triangulation consistently in classes `AIS_Shape` and `AIS_TexturedShape`. The triangulation is not cleaned in `StdPrs_ShadedShape::Tessellate()`, so the triangulation tool can process sub-shapes properly.

25147 25158	*Summary:* `OpenGl_Context` - support EGL as alternative to WGL and GLX. The portability of visualization core has been improved as `TKOpenGl` toolkit now can create `OpenGL` context using EGL on Android. This feature can be further extended to Wayland, BlackBerry and others.
25159 25227 25234 25314	*Summary:* Improvements in BVH package. The Linear Bounding Volume Hierarchy (LBVH) algorithm has been implemented as an alternative to SAH-based builders. It reduces the problem of BVH building to spatial sorting along Morton curve (or Z curve). Sorting itself is performed by using radix-sort algorithm with O(N) complexity. The algorithm produces BVH trees of lower quality compared to BVH builders but it is over an order of magnitude faster (up to 4M triangles per second).
25204	*Summary:* `PrsMgr_PresentationManager::Presentation()` - do not return the last presentation when the requested one is not found. The method `PrsMgr_PresentationManager::Presentation()` now returns NULL presentation, if the specified mode has not been found.
25213	*Summary:* `TKOpenGl` - do not use deprecated built-ins in GLSL shaders. The following modifications have been introduced to improve the behavior of GLSL shaders: ▪ New methods `OpenGl_Context::ActiveProgram()` and `::BindProgram()` have been implemented to manage the currently active GLSL program. ▪ In `OpenGl_ShaderProgram` the locations of pre-defined vertex attributes `occVertex`, `occNormal`, `occTexCoord` and `occColor` are now set up before linkage. ▪ Methods `OpenGl_ShaderProgram::Bind()`, `::BindWithVariables()` and `::Unbind()` have been removed. `OpenGl_Context::BindProgram()` should be used instead. ▪ The class `OpenGl_VertexBufferCompat` has been introduced to emulate VBO behavior on systems without VBO (compatibility with broken OpenGL drivers on Windows). ▪ `OpenGl_PrimitiveArray` now uses `OpenGl_VertexBufferCompat` when VBO is unavailable. The method `OpenGl_VertexBuffer::HasNormalAttribute()` is used to activate lighting. ▪ Methods `OpenGl_VertexBuffer::BindFixed()` and `::UnbindFixed()` have been superseded by methods `::BindAllAttributes()` and `::UnbindAllAttributes()`, which handle the active GLSL program, when it is set.
25219	*Summary:* `TKOpenGl` - disable code paths unavailable on OpenGL ES 2.0. `OpenGl` package has been revised to disable code paths unavailable on OpenGL ES 2.0 when building on Android platform or for Open GL ES.

25265	*Summary:* Visualization, `Select3D_Projector` - wrong calculation of projection line in perspective mode. The calculation of projection line has been corrected in method `Select3D_Projector::Shoot`.
25276	*Summary:* Lighting is broken if some kinds of transformation applied to a shape. The problem caused by the fact that mirror transformation breaks the OpenGL lighting completely, has been solved by inverting the order of triangle vertices in mirrored mesh (triangulation). The mesh is mirrored if its transformation matrix determinant is less than 0. The flag `Mirrored` is now stored in `OpenGl_Structure` to handle `AIS` object mirror transformations. If this flag is enabled, `glFrontFace(GL_CW)` is applied before calling draw. Additionally `GL_NORMALIZE` mode is used when scale transform is applied in order to get rid of lighting issues.
25282 25303	*Summary:* `OpenGl_PrimitiveArray` - provide built-in GLSL programs as alternative to FFP. The following modifications have been introduced to provide built-in GLSL programs as alternative to FFP (fixed-function pipeline): ▪ Built-in GLSL programs have been added in `OpenGl_ShaderManager`. ▪ New option `OpenGl_Caps::ffpEnable` allows switching between FFP and built-in GLSL programs ▪ Unused values `V3d_MULTICOLOR`, `V3d_HIDDEN` and `Visual3d_TOM_INTERP_COLOR` have been removed from enumerations `Visual3d_TypeOfModel` and `V3d_TypeOfShadingModel`. ▪ Per-pixel shading mode has been implemented in `V3d_PHONG` and `Visual3d_TOM_FRAGMENT`. ▪ `OpenGl_VertexBuffer::bindAttribute()` - normalization has been activated for `non-GL_FLOAT` types, since color attribute is defined as 32-bit vector of 4 unsigned byte values. ▪ Methods `OpenGl_Context::SetColor4fv()` and `::SetPointSize()` have been added to redirect parameters to the active GLSL program. This is an alternative to `glColor4fv()` and `glPointSize()`. ▪ New option `-shadingModel` has been added in Draw command `vrenderparams` to setup Shading Model. ▪ Unused light sources now can be dynamically disabled in a generated GLSL program.
25294	*Summary:* `PrsMgr_Presentation` - detach connected presentations on destruction. The method `PrsMgr_Presentation::Projector` has been modified to explicitly erase a presentation instead of destructing it, which led to access violations in `OpenGl_Structure::UnregisterFromAncestorStructure()`.
25306	*Summary:* TKOpenGl - support texturing within Ray-Tracing. Display of textures assigned to objects has been enabled in Ray-Tracing mode. This functionality uses OpenGL bindless textures extension and thus is available only in relatively new OpenGL drivers. If an OpenGL driver does not support this extension, the textures will be disabled.

25340	*Summary:* `AIS_InteractiveContext` - properly apply selection filters at Neutral point. `AIS_InteractiveContext::MoveTo()` has been corrected to avoid using `myMainSel->OnePicked()`.
25331	*Summary:* method `V3d_View::ConvertWithProj` returns incorrect value. Method `V3d_View::ConvertWithProj` has been corrected to return a normalized direction.
25332	*Summary:* Rewrite `MinMaxValues` methods to using of `Bnd_Box`. The methods `Visual3d_View::MinValues` and `Visual3d_View::MaxValues` now return `Bnd_Box` object instead of a set of min/max values.
25349	*Summary:* `Graphic3d_MaterialAspect::Color()` returns a value different from the one set by method `Graphic3d_MaterialAspect::SetColor`. The problem with default color of clipping plane has been fixed in `Graphic3d_MaterialAspect` class.
25358	*Summary:* `V3d_View::Reset` has incorrect condition for update. The condition of update has been corrected in method `V3d_View::Reset`.
25400	*Summary:* TKOpenGl - Structure disappears if assigned a non-default Z layer The Draw command `vpriority` has been added to test how BVH tree is updated when the display priority is changed
25403	*Summary:* Wrong storage of model units in `AIS_AngleDimension`. Model unit setters have been corrected in `AIS_Drawer`. The corresponding parameters `–modelunits`, `-showunitst` and `–dispunits` have been added to `vdimension` and `vdimparam` commands.
25417	*Summary:* `OpenGl_Context` – fix pre-processor check to load GL functions on OS X. The pre-processor check during loading of GL functions on OS X has been fixed in `OpenGl_Context` class.
25422	*Summary:* `TKOpenGl` - support re-assignment of a new window to the existing View. It has become possible to reassign a new window to the existing view reusing the same OpenGL context.
25442	*Summary:* `TKOpenGl` - prevent inclusion of system header `glxext.h`. `GLX_GLXEXT_LEGACY` has been added in `OpenGl_GlFunctions` to avoid implicit inclusion of header `GL/glxext.h` by the system header `GL/glx.h`.

Application Framework

24509	*Summary:* Suspect unused variable in `TPrsStd_ConstraintTools.cxx` The following modifications have been introduced in class `TPrsStd_ConstraintTools`: ▪ `AIS_TypeOfDist` has ceased to be used for length dimension construction. ; ▪ It has become possible to build `TPrsSrd` length constraint on one or two edges. ▪ `TPrsStd_ConstraintTool` is now allowed to build length dimension on one or two, not three, shapes. ▪ The validation of dimension plane has been added in method `TPrsStd_ConstraintTools::ComputeDistance()`.
24755	*Summary:* `TDF_Label::AddAttribute()` reverses the order of added attributes. The method `TDF_Label::AddToNode` has been modified to append newly defined attributes to the end of the attribute list rather than prepend them to its beginning to enable determinism (reproduce document contents) and facilitate debugging.
24869	*Summary:* Extend OCAF testing framework. The testing of BOP Section operations history is now supported by OCAF testing framework.
24925	*Summary:* Enable OCAF persistence without setting environment variables. Previously enabling OCAF persistence required defining some variables and supplying resource files, namely: ▪ the `Plugin` file and the `CSF_PluginDefaults` variable ▪ the `<AppFormat>` file and the `CSF_<AppFormat>Defaults` variable, where `<AppFormat>` is the value returned by `Resources()` method defined in a subclass of `TdocStd_Application`. This created inconvenience for redistribution of OCC-based end-user applications and especially libraries. The following modifications have been introduced to address this issue: ▪ The need in `FWOSPlugin` library and respective entrance in the Plugin file has been eliminated. If the entry for plugin `a148e300-5740-11d1-a904-080036aaa103` is not provided, the driver will be defaulted to `CDF_FWOSDriver` (which has been moved from `FWOSDriver` package). `FWOSPlugin` is retained, but is reduced in size (moving `FWOSDriver_Driver` into CDF). Possible next step is to remove `FWOSPlugin` completely. ▪ `TdocStd_Application` subclasses may define `Resources()` method to return an empty string, and instead directly set up `myResources` manager, which has become protected in `TdocStd_Application`. ▪ `Plugin::Load()` method can now consult `Plugin::AdditionalPluginMap()` if it cannot find the plugin using the Plugin file. This global map can be populated from the user's code. ▪ `Plugin::Load()` now accepts a Boolean parameter `theVerbose` to suppress errors sent to `std::cout`. The default value is true to preserve current behavior.

25112	*Summary:* Undo of modification of the attribute `TdataStd_IntPackedMap` works wrong. The class `TdataStd_DeltaOnModificationOfIntPack` has been improved to preserve the attribute content when a transaction is undone.
25153	*Summary:* Retrieval of selection arguments crashes the application. A check on NULL has been added into `TDF_Tool::OutReferences()` static methods.
25255	*Summary:* API to control storage with triangulation. New method `BinMNaming_NamedShapeDriver::SetWithTriangles()` allows specifying if the shape should be written with or without triangles.
25317	*Summary:* Failure on attempt to save an OCAF document with a long double array attribute in a Xml file. The commands to `Set` and `Get` array value for Integer, Real and Boolean OCAF array attributes have been implemented in `DdataStd_BasicCommands`.
25397	*Summary:* Iteration on a tree of nodes is too slow. A constant reference is now returned by `TDataStd_ChildNodeIterator::Value()` to speed up iteration on a tree of nodes.

Data Exchange

7270	*Summary:* Exception in `ShapeFix_Wireframe::FixSmallEdges`. The check if the resulting shape if not empty has been added in `ShapeFix_Wireframe` algorithm.
21802	*Summary:* Not all names are transferred from STEP to IGES via XDE. Method `IGESCAFControl_Writer::WriteNames` now iterates over all nested assemblies and collects names for shapes.
22680	*Summary:* Empty result after STEP import. The method `StepToGeom_MakeBSplineSurface::Convert()` has been improved to handle identical knots properly during STEP import.
23950	*Summary:* Names and visibility of points not saved when writing XCAF Document into STEP. The visibility and the name (located at the corresponding label) of points now can be written into the STEP file when using `STEPCAFControl_Writer`. The behavior at import is controlled by new `write.step.vertex.mode` parameter, which can be equal to: ▪ 0 - One Compound: All free vertices are united into one compound and exported in one shape definition representation (vertex name and style are lost). This mode is used by default. ▪ 1 - Single Vertex: Each vertex is exported in its own SHAPE DEFINITION REPRESENTATION (vertex name and style are not lost, but the STEP file size increases).
23951	*Summary:* Visibility of free, simple shapes is not saved when writing XCAF Document into STEP. The check for top-level shape when writing invisibility property has been removed from `STEPCAFControl_Writer::WriteColors` method.
24024	*Summary:* Slow import of specific STEP data. Bounding boxes are now used in method `ShapeFix_Face::FixOrientation` to accelerate import.
24055	*Summary:* Reading a STEP file produces invalid shape. Checks for cone-like surfaces during seam fixing have been implemented in `ShapeFix_Face::FixPeriodicDegenerated()`. Obsolete check for missing degenerated edge for cones has been removed from `ShapeFix_Face::FixMissingSeam()`.
24159	*Summary:* Colors are not imported for Step-Files created with Inventor 2014. Method `STEPConstruct_Styles::LoadStyles()` now imports `StepVisual_StyledItem` entities even if they were defined without `StepVisual_MechanicalDesignGeometricPresentationRepresentation` or `StepVisual_DraughtingModel` entity.

24990	*Summary:* `StepWrite` fails on the attached shape. A check to avoid exception has been added in method `STEPControl_ActorWrite::TransferShape.`
25010 25381 25399	*Summary:* Use `AIS_ColoredShape` for display of XCAF documents. In 3d viewer, the display of XDE documents has been implemented using the new class `AIS_ColoredShape`. The following modifications have been introduced: ▪ New methods `AIS_Drawer::HasUIsoAspect` and `AIS_Drawer::HasVIsoAspect` return true if the aspect setting `UIso` or `VIso` is active for the Drawer. ▪ `XCAFPrs_AISObject` has been converted to non-cdl class and now inherits from `AIS_ColoredShape`
25092	*Summary:* `COMPSOLIDs` are not exported to STEP. `STEPControl_ActorWrite` now can translate a Compsolid as a set of solids.
25166	*Summary:* STEP Reader - allow `GEOMETRIC_SET` entity to accept non-geometric children. `Step_ToTopoDS_Builder` has been extended to allow `GEOMETRIC_SET` entity now can accept `GeometricRepresentationItem` children
25167	*Summary:* STEP Reader - allow `NextAssemblyUsageOccurance` to accept `ProductDefinitionShape` as child instead of `ProductDefinition.` A workaround has been provided to allow `NextAssemblyUsageOccurance` to accept `ProductDefinitionShape` as child instead of `ProductDefinition`.
25168	*Summary:* STEP Writer - empty `AppliedGroupAssignment` can cause exception. Function `RWStepAP214_RWAppliedGroupAssignment::WriteStep` has been improved to correct wrong multiplicity of boundary knots during import.
25169	*Summary:* STEP Reader - allow opened shells to be an outer for `ManifoldSolidBRep.` An opened shell `ConnectedFaceSet` now can be an outer shell for `ManifoldSolidBRep`, Previously only a `ClosedShell` was required.
25333	*Summary:* Wires obtained by command `connectedges` contain internal edges. `ShapeAnalysis_FreeBounds` has been protected to avoid adding internal or external edges when the edges are connected into wires.
25405	*Summary:* STL reader does not keep shared nodes. Stl Reader has been improved to check the nodes for coincidence to avoid their duplication in the mesh. New draw command `meshinfo`, which gives the number of nodes and triangles, has been implemented.

Draw

6897	*Summary:* Impossible to change edge line type in 3D View. Commands `vsetedgetype` and `vunsetedgetype` have been implemented in `ViewerTest_ObjectCommands` to set and unset edges visibility in shading mode.
22877	*Summary:* Add plugin for VIS component. New `TKIVtk` toolkit allows creating VTK interactive view in a regular or a virtual mode (virtual windows), displaying OCCT shapes and dumping them. The corresponding DRAW plugin `TKIVtkDraw` has been added for testing VIS component. It provides the following set of commands: ▪ `ivtkinit` - creates a 3D viewer window; ▪ `ivtkdisplay` - displays named objects; ▪ `ivtkerase` - erases named or all displayed objects; ▪ `ivtkfit` - automatic zoom/panning; ▪ `ivtksetdispmode` - sets display mode for named or all displayed objects; ▪ `ivtksetselmode` - sets selection mode for named or all displayed objects; ▪ `ivtkmoveto` - imitates mouse move to input point; ▪ `ivtkselect` - imitates selection in input point.
23745	*Summary:* `ViewerText` - `vdrawtext` command should not modify global text aspect. `vdrawtext` command has been improved to avoid modifying global text aspect.
23828 24589	*Summary:* Disable floating-point exceptions by default. The following modifications have been introduced to disable floating-point exceptions by default: ▪ `OSD::SetSignal()` is now called with `Standard_False` argument to disable FPE. ▪ 0.0 / 0.0 uncertainty in circle to quasi-angular bspline conversion has been fixed in method `Convert_ConicToBSplineCurve::BuildCosAndSin.`
24202	*Summary:* Support class methods as callbacks for Draw Harness commands. The interface of `Draw_Interpretor` class has been extended by the possibility to register a new DRAW command implemented as method of a class (rather than plain C function).
24430	*Summary:* Command `vviewlist` does not add the name of viewer created by command `XShow`. The class `ViewerTest_Tool` has been removed. The unified way to create a viewer is using `ViewerTest::ViewerInit` method.

24644	*Summary:* `Draw_Printer` - provide the way to control messages gravity filter. The unified interface to setup message gravity filter has been provided by methods `SetTraceLevel()` and `GetTraceLevel()` from `Message_Printer` class. New Draw command `dtracelevel` has been implemented to show/change message gravity filter of printers registered for the default messenger Additionally, Draw printer has been defined in `TKDraw` instead of `TKXSDRAW`.
24688	*Summary:* Command `vdump` makes snapshot only for the first view. The command `vdump` has been adapted to the multi-view functionality.
24705	*Summary:* Add `–noupdate` option to `vdisplay` command. The option to avoid viewer redraw has been added to commands `vdisplay`, `vdonly`, `verase` and `vremove`.
24712	*Summary:* Extend DRAW commands `vertex` and `extrema`. DRAW command `vertex` from `BRepTest_CurveCommands` has been extended to create a vertex from point. DRAW command `extrema` from `GeometryTest_APICommands` has been extended and unified: both extrema points are output to DRAW if they are too close.
24757	*Summary:* Move commands `fixshape`, `tolerance`, and similar to Modeling Algorithms. ▪ DRAW commands from package `SWDRAW` related to Shape Healing component have been moved to the appropriate `TKTopTest` toolkit in Modeling Algorithms. ▪ Class `SWDRAW_ToVRML` has been moved to `XSDRAWSTLVRML_ToVRML`. ▪ Unused classes `SWDRAW_ShapeBuild`, `SWDRAW_ShapeConstruct` and `XSDRAW_Shape` have been removed. ▪ Unused commands `NSPApply` and `xtrace` have been removed as well.
24765	*Summary:* Compile errors on Mac OS X with GCC 4.2. Sequence has been replaced by handle in map, to avoid involvement of copy constructor (private) in `ViewerTest_ViewerCommands` and `ViewerTest` classes.
24802	*Summary:* `Help` command should not apply implicit rules to define what command description it should show. Now the expression `help <command>` displays only the information about `<command>`. To get the information about all commands starting with `<command>`, use expression help `<command>*`.
24873	*Summary:* `ViewerTest` - add command `vbounding` to show presentation bounding box. New command `vbounding` has been added to show the bounding box of the presentation

24961	*Summary:* MeshVS: revision of DRAW commands. DRAW commands for testing MeshVS package have been revised and some new commands have been added to cover advanced MeshVS features, such as visualization of vector fields and text labels, 3D mesh elements support and deformed mesh visualization. The following Draw commands have been improved: ▪ The default mesh link color used by meshfromstl command has been changed from white to yellow to avoid ambiguity with the selection color. ▪ meshfromstl now displays MeshVS_Mesh object consistently with vdisplay command; ▪ meshdel command has been deleted; vremove can be used instead; ▪ meshshowall now sets empty maps of hidden IDs instead of null handles; ▪ meshhidesel now initializes the maps of hidden IDs to avoid exception; ▪ Optional transparency argument has been added in meshmat command. Transparency has been enabled in the viewer. ▪ New command meshvectors has been added to test VectorPrsBuilder.
25016	*Summary:* Command splitshape does not split attached face by attached edge. Method LocOpe_WiresOnShape::FindInternalIntersections has been corrected to provide correct work of splitshape command.
25018	*Summary:* Misprint in argument -settransparancy for command vaspects. A misprint in argument -settransparancy for command vaspects has been fixed.
25019	*Summary:* Command bsection builds a pcurve on the second shape slowly. The approximation flags have been changed according to the order of faces in method IntTools_FaceFace::Perform.
25020	*Summary:* checkoverlapedges does not check for degenerated edges. The presence of degenerate edges is now checked in command checkoverlapedges.
25198	*Summary:* Add -noupdate option to vsetlocation command The option –noupdate, has been implemented for vsetlocation command. This option prevents the view from being redrawn and can be useful to assign locations for many objects at once.
25217	*Summary:* Activate GLSL warnings output within command vgldebug. GLSL warnings output within command vgldebug has been enabled.

25229	*Summary:* Capping has grey tint if the color is set with command `vclipplane`. `Graphic3d_TypeOfMaterial::SetColor()` has been replaced with `Graphic3d_TypeOfMaterial::SetAmbientColor()` and `Graphic3d_TypeOfMaterial::SetDiffuseColor()` in method `ViewerTest_ViewerCommands::ClipPlane()`.
25231	*Summary:* Improvement of Draw top level menu. The following improvements have been introduced in the layout of Draw Test Harness application: ▪ The items from "Command" menu have been provided with meaningful texts, with names of corresponding DRAW commands in parentheses. ▪ New menu "Help" provides access to three items: "User Guide", providing a short-cut to Draw User's Guide; "Commands", providing access to information accessible via command `help` and "About", showing OCC logo, copyright statement, and a link to OCC web sites. ▪ New menu "Samples" allows running samples located in `samples/tcl`. This menu is populated automatically by available scripts.
25235	*Summary:* Improve commands `vdimension` and `vdimparam`. The following new options have been added to `vdimension` and `vdimparam` commands: ▪ `textmode` – sets 3D text display mode; ▪ `textsize` - sets 3D text size; ▪ `arlength` and `arangle` change arrow parameters; ▪ `color` – sets 3D text color. Additionally the initialization of angle dimension has been corrected to allow ellipse input geometry with equal radiuses.
25267	*Summary:* Implementation of `callback` before DRAW exit. `Callback` mechanism has been implemented in `Draw_Window`.
25269	*Summary:* Make parallel version of Boolean operations available from DRAW. New command `bparallelmode [1/0]` has been added to enable or disable the parallel mode for Boolean operations. Sequential computing is used by default.
25289	*Summary:* `ViewerTest` - Support RGB color definition as input for `vsetcolor` and `vaspects` commands. RGB color definition has been provided for commands `vsetcolor` and `vaspects`. Additionally, method `Quantity_Color::ColorFromName` has been added to find color from predefined names.
25291	*Summary:* `ViewerTest` - replace command `vchangecamera` by `vcamera`. ▪ The syntax of command `vcamera` has been improved. ▪ The occurrences of `atof()` have been replaced by `Draw::Atof()` in `ViewerTest_ViewerCommands`.

25316	*Summary:* Display or erase objects in the local selection context. New option `-local` has been added in commands `vdisplay`, `vdisplayall`, `verase`, `veraseall` and `vremove` to execute these commands only on the objects in the local selection context.
25344	*Summary:* Draw variables do not follow the scope of TCL level. `NCollection_DataMap` has been implemented instead of `TCollection` in `Draw_VariableCommands.cxx` to restrict by the scope of a procedure the life time of Draw variables created inside it. `StandardCommands.tcl` has been fixed to allow Draw commands to treat variables declared by `upvar` command correctly.
25359	*Summary:* Content of menu-items "Curves" and "Surfaces" is increased after each sample launching. The menu items "Curves" and "Surfaces" are now destroyed before displaying in Draw Test Harness application:
25383	*Summary:* Broken command `vrecord`. The command `vrecord start` has been fixed to pass HWND of the current 3D view to `OpenGl_AVIWriter::StartRecording()`.
25410	*Summary:* Tool for extended validity check of curve on surface. The following Draw commands have been implemented to check the validity of a curve on surface: ▪ `xdistef <edge> <face>` computes the distance between points of a 3D curve <edge> and the corresponding points of its 2D projection on <face>. The checking algorithm is implemented in method `BOPTools_AlgoTools::ComputeTolerance`, which computes the maximum distance between points taken from 3D and 2D curves ▪ `checkcurveonsurf <shape>` checks each edge/face pair in the shape using `ComputeTolerance` method and stores invalid pairs to `myResults`. The pair is invalid if the tolerance value of the edge is less than the value computed by `ComputeTolerance` method. The command `checkcurveonsurf` also gives suggestions to fix the shape by increasing tolerance values of the invalid edges. In some cases the tolerance values suggested by the tool can be very large. Such values should be used very carefully, because setting large tolerance values to the sub-shapes of a shape can make it non valid (self-interfered) or lead to an unexpected result when it is used in some operations.
25445	*Summary:* Command `incmesh` should support all parameters used in `BRepMesh`. Support for angular deflection and relative flag arguments has been introduced in command `incmesh`.

Mesh

23106 25157 25412 25293	*Summary:* `BRepMesh_IncrementalMesh` returns wrong status. The following changes have been made to improve `BRepMesh_IncrementalMesh` algorithm: ▪ Storing of intermediate structures to the shapes in `BRepMesh_FastDiscret` and their restoring in `BRepMesh_FastDiscretFace` has been eliminated. Common data structures have been moved to `BRepMesh_FaceAttribute`; ▪ The procedures that fill data structures using existing and newly created triangulation have been unified. The corresponding tools are now provided by `BRepMesh_EdgeTessellationExtractor` and `BRepMesh_EdgeTessellator` classes; ▪ Code duplications have been removed from `BRepMesh_FastDiscret` and `BRepMesh_FastDiscretFace`; ▪ Mutexes have been removed from `BRepMesh_FastDiscret` and `BRepMesh_FastDiscretFace` as data structures are divided into shared and local ones; ▪ Resulting triangulation is now stored within a single place in `BRepMesh_IncrementalMesh` class; ▪ A redundant flag `isPositive` has been removed from `BRepMesh_Delaun` class because the face is always oriented forward.
23631	*Summary:* Infinite memory consumption in `BRepMesh.` New classes `BRepMesh_FaceChecker` and `BRepMesh_EdgeChecker` have been implemented to check if the meshed shape has correct polygon data, i.e. if `PolygonOnTriangulation` of a particular edge is connected to the same Triangulation data structure as stored inside its parent face. `BRepMesh_IncrementalMesh` class is now imported from `BRepMesh` package.
24219	*Summary:* Strange if-statement in `BrepMesh_IncrementalMesh.` The if-statement has been simplified in `Update()` methods from `BrepMesh_IncrementalMesh` class.
24968	*Summary:* Improve `BRepMesh_Classifier` to cope with intersection of huge number of wires. Two-pass approach for intersection check with possibility to run it in parallel mode has been implemented in `BRepMesh_Classifier`: ▪ At the first pass - bounding boxes of segments are checked for overlapping; ▪ At the second pass – the intersection point is calculated if the overlapping is detected.

25039 25088	*Summary:* Improvement of code structure of general and supporting tools implemented in `BRepMesh`. The following changes have been made to improve the code structure of general and supporting tools in `BRepMesh` package: ▪ CDL declarations and LXX parts of classes have been replaced by usual header files; ▪ Data containers have been replaced by their equivalents from `NCollection`; common data types are declared in `BRepMesh_Collections.hxx` within `BRepMeshCol` namespace; ▪ Redundant class `BRepMesh` and static method `Mesh` have been removed; ▪ The most transparent approach has been implemented to calculate a circle around a triangle in `BRepMesh_CircleTool` class; ▪ Static method `BRepMesh_GeomTool::IntLinLin` has been added to eliminate code duplication in `BRepMesh_Dealun` and `BRepMesh_CircleTool` classes; ▪ Function `BRepMesh_Write` storing `BRepMesh_DataStructureOfDelaun` to a `BRep` file has been introduced for debug needs; ▪ Method `BRepMesh_DataStructureOfDelaun::ElementNodes` has been introduced to retrieve ordered nodes forming the given triangle. The following changes have been introduced in the API of `BRepMesh` and StdPrs packages: ▪ Methods `BRepMesh_FastDiscret::BoxMaxDimension` and `::RelativeEdgeDeflection` have been moved to `BRepMesh_ShapeTool`; ▪ Method `BRepMesh_FastDiscretFace::FindUV` has been moved to `BRepMesh_ShapeTool`; ▪ `BRepMesh_Circ` has been renamed into `BRepMesh_Circle`; ▪ Some methods from classes `BRepMesh_DataStructureOfDelaun` and `BRepMesh_CircleTool` have been renamed for consistency reasons; ▪ The common part of `IntSegSeg` has been moved from `BRepMesh_Delaun` to `BRepMesh_GeomTool` as a static method; ▪ Redundant code for iteration among faces of shape/edges of face/vertices of edge has been removed from `BRepMesh_ShapeTool`. The usual `TopExp_Explorer` tool can be used instead. Method `BRepMesh_ShapeTool::Orientation` can be replaced by the code querying `Orientation()` parameter directly from `TopoDS` shape. ▪ Dependency from `BRepMesh_ShapeTool` has been removed from `StdPrs_ToolShadedShape`. ▪ Now `StdPrs_ToolShadedShape` provides only three static methods `IsClosed`, `Triangulation` and `Normal`. The corresponding changes have been made in packages `DBRep`, `Voxel`, `MeshTest`, `StdPrs`, `StdSelect`, `StlTransfer`, `VrmlConverter`, `VrmlData` and `XSDRAWSTL` according to the changed API of `BRepMesh_IncrementalMesh`.
25179	*Summary:* `BRepMesh` cannot build mesh on a specific geometry. It is now checked that the medium point does not coincide with the end points in method `GCPnts_TangentialDeflection::PerformCurve`.
25310 25320 25364	*Summary:* Keeping temporary data structures in `BRepMesh_FaceAttribute` leads to out-of-memory exception. Memory management in `BRepMesh_FastDiscret` class has been changed to immediately remove unused face attributes.

Shape Healing

24218	*Summary:* `ShapeFix_Face` requires double execution to produce valid shape when `FixSplitFaceMode` is in effect. Check of wire orientation before adding an internal wire in the splitting face has been implemented in method `ShapeFix_Face::FixSplitFace`.
24249	*Summary:* Crash on `ShapeFix_Shape.` The boundaries of "natural bound addition" in command `FixAddNaturalBound` have been restricted: a face that does not contain an outer wire, should not have any infinite UV boundaries because building a new face using a surface requires specified UV boundaries. Additionally, when `FixAddNaturalBound` creates a new face with natural boundary, `myResult` is now updated for the next "fix small-area wires" algorithm In `IsPeriodicConicalLoop` command, incorrectly working `BRepTools_WireExplorer` has been replaced by `TopoDS_Iterator` and a natural bound is added to all faces constructed with UV-periodical surfaces (not only sphere and torus; e.g., closed b-splines).
24958	*Summary:* Numerous `ShapeFix_IntersectionTool` code fixes. Periodic normalization has been optimized in method `Geom_BSplineSurface::PeriodicNormalization`. The intersecting edge is not split but truncated if one of split parts is too small.
24960	*Summary:* Value of result tolerance after `FixShape` is bigger than value of set "Maximal tolerance". The description of how to use "Maximal tolerance" and "Minimal tolerance" values in class `ShapeFix_Shape` has been updated.
24983	*Summary:* For the incorrect seam edge in STEP file no fix is provided. The class `ShapeFix_Face` has been improved to fix notched edges twice per face: before and after checking for a missing seam. The check for invalid Surface of linear extrusion (with basis curve-line and extrusion direction parallel to this line) during STEP surface conversion has been added in class `StepToGeom_MakeSurfaceOfLinearExtrusion`.
25012	*Summary:* `ShapeFix_Wire` checks a wrong parameter for the curve during `SameParameter` check. `SameParameter` check in method `ShapeFix_Wire::FixEdgeCurves()` has been fixed to verify correct pairs of parameters.
25122	*Summary:* Expose internal function `ShapeAnalysis_Edge::ComputeDeviation()` `ShapeAnalysis_Edge::ComputeDeviation()` has become a static method (earlier it was an internal unction). This method computes the maximal deviation between two curve representations.

Configuration

24718	*Summary:* Broken build on VC8 (VS 2005) if TBB is used. Macro `_WIN32_WINNT` has been defined in `OpenGl_SceneGeometry.cxx` for building with TBB on VS 2005.
24858	*Summary:* Convert class `V3d_ListOfTransient` to non-CDL. Class `V3d_ListOfTransient` has been converted to non-CDL (pure `HXX`) form.
24888 25140 25141 25244	*Summary:* Refactoring of OCCT CMake meta-project. The following improvements have been introduced in OCCT CMake meta-project: ▪ The user now can see only directories of libraries, shared libraries and headers of 3rd-party products. ▪ 3rd-party search priority has been changed to look preferably for user paths and only as a second choice for system paths ▪ OCCT information, generated by WOK, has been divided in two files: `occt_toolkits.cmake` and `occt_inc_toolkits.cmake`. ▪ The search algorithm has been adapted for Freetype 2.5.1 and above and now properly finds `config/ftheader.h`. ▪ Tcl, Freetype and VTK are now checked before usage of CSF variables ▪ The search for Freetype and tcl now can be handled by default CMake mechanism. ▪ The macro `3rdparty` has been added to find debug libraries (and release if debug ones are not found); ▪ Variables with suffix `_DLL` are now used only in window case. ▪ `-DDEBUG` flag has been removed from `CMAKE_CXX_FLAGS_DEBUG` and `CMAKE_C_FLAGS_DEBUG`.
24907	*Summary:* Compilation fails on vc12 (VS2013). `#include <algorithm>` has been added in `BVH_BinnedBuilder`, `Image_AlienPixMap` and `Image_PixMap` to avoid compilation problems.
24989	*Summary:* Workaround for VS12 32-bit compiler causes crash in `AIS_Axis`. The method `AIS_Axis::ComputeFields()` has been corrected to avoid crash at the attempt to create an `AIS_Trihedron` instance in optimized mode with `/O2` compiler option.
25051	*Summary:* Compilation issues on OS X 10.6.8. `OpenGl_Window_1.mm` and `OpenGl_Context_1.mm` have been corrected to prevent inclusion of system `glext.h` on Mac OS X 10.6.8.
25161	*Summary:* Exception in Draw trying to upload ALL (especially `TOPTEST`). `SWDRAW.cxx` has been modified to avoid exception.

25215 25438	*Summary:* Porting to Android - fix minor issues. The following issues have been corrected for OCCT porting to Android: ▪ `STEPConstruct_AP203Context = pw_gecos` member of password structure has been disabled in Android case. ▪ Usage of `_atomic_inc()/_atomic_dec()` has been corrected in `Standard_Atomic`. ▪ The use of `/dev/zero` and `/dev/null` for allocation of memory is now avoided in `Standard_MMgrOpt`. ▪ In OpenGl, viewport is now always set up within `OpenGl_Workspace::Redraw()`.
25247	*Summary:* Ensure correct end-of-line in test scripts. File `.gitattributes` has been modified to provide correct end-of-line in test scripts.

Samples

24548	*Summary:* Ray Tracing mode not available in MFC samples. Support for ray tracing rendering has been implemented in MFC samples similarly to Qt samples (only in samples using 3d Viewers).
24699	*Summary:* Prototype interoperation of TKOpenGl viewer with Direct3D viewer. WPF, a new CSharp sample, has been implemented to provide rendering of the OCCT scene to a Direct3D context. DirectX SDK is required to build this sample.
24758	*Summary:* Sample DRAW scripts for demonstrating sweeping algorithm. Two scripts creating complex shapes, which can be produced by sweeping, have been added to demonstrate the capabilities of sweeping algorithm in OCCT: ▪ `samples/tcl/drill.tcl`: creates a model of a twist drill; ▪ `samples/tcl/cutter.tcl`: creates a simplified model of a milling cutter.
24943	*Summary:* Port MFC sample to UNICODE for compatibility with VS2013. MFC samples have been ported to UNICODE to provide compatibility with VS2013: • VC12 project files have been added for MFC samples; • `Unicode` option has been added for MFC samples in CMake; • `MFC` option is not set globally in CMake; • The description of CMake building procedure for MFC sample has been updated.
25031	*Summary:* Incorrect initialization function call. Incorrect function call has been removed in `Qt` samples.

25149 25195	*Summary:* Add Java and Qt5/QML samples for Android 4.x. Dedicated samples based on Java and Qt5/QML have been implemented for Android platform. Their functionality is similar to MFC-based Import-Export sample.
25236	*Summary:* Sample DRAW script to demonstrate XDE features. New sample `samples/tcl/xde.tcl` has been added to show the possibilities of Extended Data Exchange component.
25284	*Summary:* Problems with standard MFC samples. The following improvements have been made in MFC samples: • Conversion from `LPCTSTR` to `Standard_CString` has been corrected in Save Document option of OCAF sample; • Mouse move event handling for HLR 2D view as well as display and update of `CSelectionDialog` view have been corrected in HLR sample; • Code duplication is avoided `CGeometryDoc` from Geometry sample; • `OCC_3dBaseDoc::DragEvent()` handler now emulates rectangle selection properly; • Resource files have been moved from `/src` to `/res` in Viewer3D sample.
25290	*Summary:* `.gitignore` - do not track generated files with extensions `aps`, `opensdf` and `ipch`. Junk files have been removed from MFC samples.
25327	*Summary:* Operation `UNDO` does not remove previous presentation in Qt CAD Assistant. Qt CAD Assistant sample has been fixed to provide correct `UNDO` for Explode operation.
25361	*Summary:* `.`Sample Voxel crashes during Demo Collisions. The definition of array has been improved in method `Voxel_CollisionDetection::Clear()` to avoid demo sample crash.
25387	*Summary:* Error appears during export to STEP operation in `ImportExport` MFC sample. The combo-box for selection of triangulation mode has become available in “Export to STEP and CSFDB files” dialog. By default the selection is automatic.

Coding

20716	*Summary:* Eliminate usage of `config.h` header file. OCCT has been revised to eliminate all occurrences of `config.h`, including macros defined by it.
24240	*Summary:* Separation of header files for `QANCollection` into `hxx` and `cxx`. Files in `QANCollection` package have been reorganized to avoid placing code in header files.
24444	*Summary:* Fix compilation issues on some not fully POSIX compliant Unix systems. The packages `Standard`, `OSD`, `Aspect` and `Xw` have been revised to fix compilation issues.
24552	*Summary:* Conversion of the generic classes to the non-generic (`BndLib`). Methods `BndLib_Compute2d::Compute()` and `BndLib_Compute3d::Compute()` have been replaced by the template method `BndLib::Compute()`.
24660 24661 24662 24663	*Summary:* Remove unused generic classes. See the list of removed generic classes and instances in the Appendix.
24545 24547 24683 24708 24727 24761 24763 24773 24774 24778 24885	*Summary:* Conversion of generic classes into non-generic. See the list of removed and renamed generic classes and instances in the Appendix. Other changes: ▪ Classes `math_Vector` and `math_IntegerVector` have become non-cdl. ▪ Packages `GccGeo`, `GraphDS`, `Dynamic`, `Materials` and `TKAdvTools` containing unused generic classes have been removed from toolkit `TKAdvTools`. ▪ Package `GraphTools` has been moved to WOK. ▪ Packages `Expr` and `ExprIntrp` have been moved to `TKMath`. ▪ Internal classes `TFunction` and `Ufunction` have been moved from package `Gprop` to package `Geom2dHatch`. ▪ Internal classes `FunctionTanCuCuOnCu`, `FunctionTanCuCuCu`, `FunctionTanCirCu`, `GccIter_FunctionTanCuCu`, `FunctionTanCuPnt`, and `FunctionTanObl` and enumerations `Type1`, `Type2` and `Type3` have been moved from package `GccIter` to package `Geom2dGcc`.
24634 25038	*Summary:* Remove unused classes from package `Aspect`. `Aspect_FontStyle`, `Aspect_FontMap`, `Aspect_WidthMap`, `Aspect_TypeMap`, `Aspect_MarkMap`, `Aspect_Driver` and other unused classes have been removed from package `Aspect`.
24710	*Summary:* Avoid including `Handle_Class.hxx` headers wherever it is not necessary. The OCCT has been revised to replace includes of `Handle_Class.hxx` headers by `include` of the corresponding class wherever it is possible without redesign of the classes.

24737	*Summary:* Remove `br` tags from header files. The OCCT has been revised to remove redundant occurrences of tags `br` and `#ifdefs` around `#includes`.
24742	*Summary:* Remove rarely used collection classes. The following classes defined in `TCollection` and `NCollection` packages and giving no benefit over widely used collections such as `List` and `Sequence` have been removed. Other changes: ▪ In `Adaptor3d_CurveOnSurface`, calculation of continuity intervals has been refactored to build and store a sorted sequence of reals, instead of collecting them to set, copying to array, and then sorting. ▪ Generic class `TCollection_Slist` and nested class `TCollection_SlistNode` have been replaced by `TopLoc_SlistOfItemLocation` and `TopLoc_SlistNodeOfItemLocation`.
24784	*Summary:* Move documentation in CDL files to proper location. OCCT has been overhauled and many CDL files have been modified to import comments into generated `hxx` files correctly.
24792	*Summary:* Remove unused hacks for compilers without STL. Unused macro definition `USE_STL_STREAM` that has a compiler-provided analogue has been removed.
24794	*Summary:* Drop unused files `Standard_osd_ffi.c` and `Standard_symlink.c`. Unused files `Standard_osd_ffi.c` and `Standard_symlink.c` have been removed.
24804	*Summary:* `OSD_PerfMeter` documentation is broken. The comments for class `OSD_PerfMeter` have been corrected.
24805	*Summary:* Eliminate unused static functions and methods: `ShallowDump()`, `ShallowCopy()`, `STANDARD_TYPE()`. The following obsolete static functions and methods have been removed: ▪ Implementation of global functions `STANDARD_TYPE()` for types not inheriting `Standard_Transient` or `Standard_Persistent`; ▪ Global functions and class methods `ShallowCopy()`; ▪ Classes `Visual3d_PickPath` and `Visual3d_PickDescriptor`; ▪ Global functions and class methods `ShallowDump()` except for classes `Standard_GUID`, `TopLoc_Datum` and `TopLoc_Location`, which are still used in some Debug printouts.
24814	*Summary:* Avoid using explicit names of `Handle` classes. The code has been revised to refer to `Handle` classes using macro `Handle(T)`, which expands to `Handle_T`. Previously some expanded names were used explicitly, which caused problems when macro `Handle()` was changed.

24830	*Summary:* Remove redundant keyword 'mutable' in CDL declarations. Keyword mutable has been removed from CDL files in cases when it is not required and even causes errors in CDL extraction if the corresponding type (returned or argument) is converted to non-CDL class: ▪ in return statements, such as `returns mutable`… ▪ in function argument declarations, such as `Method (me; var: mutable Integer)`;
24841	*Summary:* Remove duplicated and/or unused instances of collections. ▪ Inheritance of `MeshVS_ColorHasher` from `TColStd_MapIntegerHasher` has been dropped as it leads to problems if `TColStd_MapIntegerHasher` is converted to template. ▪ The first of two overloaded functions `Draw::Load()` has been removed as unused.
24847 24862 24863 24893	*Summary:* Clang warnings. The following modifications have been introduced to get rid of Clang warnings: ▪ Obsolete `OpenGl_Display` class has been removed to eliminate `–Wdeprecated-writable-strings` warning; ▪ Clang warning `–Wint-to-pointer-cast` has been fixed in class `TestTopOpeTools_Trace.` ▪ Incorrect comparison has been fixed in conditions within `RiseIf` macro in `GC`, `GCE2D` and `gce` packages to eliminate `–Wlogical-not-parentheses` warning. ▪ Class `Draw_Vmap` has been removed. Now the objects are stored using `Ncollection_DataMap`. `Tcl` variable is associated with the object using its name.
24892	*Summary:* `AIS_TexturedShape::ShowTriangles` overloads mismatch. The parameter `theToShowTriangles` in one of methods `AIS_TexturedShape::ShowTriangles` has become non-optional to avoid confusion.
24987	*Summary:* Add missing implementation of `Select3D_SensitiveTriangulation::DetectedTriangle().` Inline method `Select3D_SensitiveTriangulation::DetectedTriangle()` has been properly implemented.
24912	*Summary:* Remove unused `Graphic3d_Strips.` Unused class `Graphic3d_Strips` has been removed.
24936	*Summary:* Remove unused package `IncludeLibrary.` Unused package `IncludeLibrary` and file `IGESToBRep/project.pxx` have been removed.

24937	*Summary:* New compile warning in `Extrema_GenExtCC.gxx` on MacOS. The keyword `redefined` has been added to the definitions of `delete` function in classes `math_MultipleVarFunctionWithGradient`, `math_MultipleVarFunction` and `math_MultipleVarFunctionWithHessian`.
24941	*Summary:* Need whitespace between adjacent string literals for Clang in C++11 mode. A whitespace has been added in files `Interface_Version.hxx` and `STEPControl_Controller.cxx` between macro `OCC_VERSION_STRING` and string literals to avoid Clang warning.
24955	*Summary:* Misuse of `DownCast`. `DownCast` has ceased to be applied to `Handle` of the type, which is not a base class of the target.
24972	*Summary:* Influence of the order of includes during compilation. OCCT has been revised to include necessary files in some cases that caused compilation problems. Unused files have been removed from package `PColStd`.
24974	*Summary:* GCC compiler warnings `backslash inside comment`. Comments have been corrected in files `MAT2d_MiniPath.cdl` and `IntCurveSurface.cdl` to avoid GCC warnings.
24992	*Summary:* Missing implementation. The method `GeomConvert_CompCurveToBSplineCurve::Clear()` has been properly implemented
25026	*Summary:* `Graphic3d_Plotter` – remove unused class. Unused class `Graphic3d_Plotter` and methods related to plotter support in other classes, such as `Visual3d_View`, `V3d_View`, `Graphic3d_Structure`, `Aspect` and `Graphic3d` have been removed.
25074	*Summary:* `Intf_Polygon2d` – add virtual destructor. Virtual destructor has been added in method `Intf_Polygon2d::Closed` to avoid compilation warnings
25077	*Summary:* Explicit initialization of copy constructor. `Standard_Failure` class has been improved to add explicit calls of copy constructor of the base class, where compiler warnings are reported.
25090	*Summary:* Drop `Graphic3d_Group::MyIsEmpty` flag. The redundant flag `Graphic3d_Group::MyIsEmpty` has been removed.

25094	*Summary:* Code clean up in Visualization and `Visual3d_View`. Classes `Visual3d_ViewManager` and `Visual3d_View` have been revised to eliminate debugging sections, impossible conditions, and obsolete solutions escaped by macros.
25154	*Summary:* Collections in `BRepMesh` package are named in a non-conformant manner. `BRepMesh` package has been converted to nocdlpack. Its collections have been defined in `BRepMesh` namespace. Thus the package has become compatible with wrapping tools, such as SWIG.
25182	*Summary:* `Standard_OVERRIDE` – add alias for C++11 `override` modifier. An alias for C++11 `override` modifier has been added in classes `AIS_ColoredShape` and `Standard_Macro`.
25216	*Summary:* `WNT_Window` – remove unused fields and types. Deprecated constructor `WNT_Window` that took two 16-bit numbers, which could be dangerous on 64-bit systems, has been removed. Other redundant classes have been removed from `InterfaceGraphic` package and `WNT`.
25266 25418	*Summary:* Debug statements in the source are getting flushed on to the console. Throughout OCCT, all macros having names with `DEB` at the end have been replaced with macros `OCCT_DEBUG`. The names of macros that start with `DEB` have been modified to start with `OCCT_DEBUG_`.

Documentation

24117	*Summary:* Enhance documentation of `BRepMesh_IncrementalMesh`. `BRepMesh_IncrementalMesh` algorithm has been described in Modeling Algorithms User's Guide.
24649	*Summary*: New user's guide for Boolean operations. New user's guide explaining how Boolean operations work has been added to the documentation.
24854	*Summary:* Merge of Technical Overview from opencascade.org site into the documentation. Some information and images that were previously available only on `opencascade.org` site have been merged into `technical_overview.md`.
24916	*Summary:* Use custom-built Tcl instead of `ActiveTcl` distribution Instructions for building Tcl/Tk from sources have been added in the User's documentation.
25072	*Summary:* Environment variable `MMGT_REENTRANT` is still mentioned in the Overview although it was removed. References to `MMGT_REENTRANT` have been removed from the documentation.
25205	*Summary:* `BRepOffsetAPI_MakeFilling` algorithm documentation is insufficient. The comments about `BRepOffsetAPI_MakeFilling` algorithm have been added in `BRepOffsetAPI_MakeFilling.cdl` and `BRepFill_Filling.cdl`.
25226	*Summary:* Reference Manual – list of classes in package is polluted. The method generating reference documentation has been corrected to avoid adding irrelevant classes to the package list.
25343	*Summary:* Update user's guide and DRAW commands description. Visualization User's Guide has been revised and updated with features recently introduced in the component, such as stereographic rendering, shading, ray tracing support, etc.
25353	*Summary:* Link to tcl broken. A broken link has been fixed in overview.md.
25443	*Summary:* Update packages names in Building 3rd-party libraries on Linux. Instructions on building 3rd party tools and readme files for samples have been updated.

WOK

16488 24557	*Summary:* Improve formatting of HXX files generated by WOK. The following improvements have been introduced in the formatting of header files generated by WOK from CDL files: ▪ Wrapping each `#include` statement into `#ifdef` is avoided. ▪ A newline is added after each method declaration
24713	*Summary:* WOK compilation fails on tcl8.6. WOK now can be compiled with `tcl8.6`.
24724	*Summary:* NT file routines in WOK use `OSD_File::myFileChannel` instead of `myFileHandle`. WOK has become consistent with the new implementation of `OSD_FileNode` and `OSD_File`.
24740	*Summary:* WOK headers generation – improve syntax. The output of WOK headers generation has become more conform to OCCT coding rules (redundant spaces removed, spaces added in the arguments list).
24769	*Summary:* Extend CDL syntax with “imported transient class” keyword. Classes inherited from `Standard_Transient` are now imported as transient and managed as `Handle(Package_ImportedClass)` in arguments and class fields. Previously all imported classes were processed as non-transient ones.
24815	*Summary:* `msvc.bat` uses wrong syntax for `start.exe`. A misprint in `msvc.bat` has been corrected.
24963	*Summary:* Problem with generation VS projects for Microsoft Visual Studio 2013 by WOK. A misprint in `template.vc12` and `template.vc12x` has been corrected.
24993	*Summary:* Adding VTK support to WOK. The possibility to add VTK to prerequisites is now supported by WOK.
25156	*Summary:* Fix compatibility with Emacs 24. WOK has become compatible with Emacs 24. Emacs functions `set-screen-width` and `set-screen-height` now can be read properly.

Products

CAM Library

23406	*Summary:* Create functions to build the shortest geodesic path between two points on a shell. The data structure for ordering windows by minimal depth has been optimized in the method `Geodesic_MeshDistances::Compute()`.
25143	*Summary:* Extend and improve the class `Geodesic_MeshDistances`. The following improvements have been introduced: ▪ New method `Geodesic_MeshDistances::SetSource` has been implemented to set the seed defined by a point and links of a triangle inside which the point is located. Information about links containing the shortest path points has been provided; ▪ The ability to use a point already projected on a mesh to build the distance field has been implemented in the method `Geodesic_ShellDistances::BuildPath`

Express Mesh

22984	*Summary:* Express Mesh produces very narrow triangles at face boundaries. A Delaunay-based algorithm has been implemented in `QMBgr_FacetBuilder` providing protection against skinny triangles. New option –`delaunay` has been added to `EmeshTest` to enable additional check for Delaunay condition.
23503 25126	*Summary:* Incorrect processing of faces with internal wires or edges. The following improvements have been introduced in the meshing algorithm: ▪ The processing of open inner wires consisting of internal edges has been implemented in `QMTools_WireCorrector` class; ▪ `QMTools_WireIterator` has been protected against exception due to manipulations on empty list of edges; ▪ `QMShape_Tessellator::DiscretiseWire` has been protected against `OutOfRange` exception produced by `ShapeAnalysis_WireOrder` due to disabled processing of non-manifold wire topology (internal edges); ▪ `QMData_Face` has been protected against exception due to void bounding box of wire with an empty list of edges; ▪ `QMShape_Tessellator::Perform` now checks that a wire does not contain internal or external edges.

24408	*Summary:* Provide option to generate quad mesh. The algorithm that generates a quad-only mesh for a `TopoDS_Shape` shape has been enabled as a special mode. To use it, define `SetMeshElemType` as `Quad`. Since `TopoDS_Shape` can store only triangular meshes, additional pure virtual interface `QMData_IMeshFace` is used to store the produced quad mesh. The command `MFmeshshape` has been added to the OMF component to test the algorithm and to output its results to an OMF mesh object.
24419	*Summary:* Express Mesh fails on two simple faces. `QMData_MeshParameters` and `QMShape_Tesselator` have been modified to extract indices of bad shapes directly from `Tesselator` class. The old approach (checking a discretized model) has become obsolete as a data model is emptied at the end of meshing.
25007	*Summary:* Express Mesh hangs on a simplistic case. The default deflection value in Express Mesh is now equal to 0.001 instead of 1e-7.
25079	*Summary:* Wrong declaration of RTTI. The declaration of RTTI has been corrected in `QMData_Imesh.hxx`.
25097	*Summary:* Express Mesh fails to mesh two simple faces restored from binary format. The use of pointer to reference has been changed in classes `QMShape_DiscrCurve` and `QMShape_Tesselator`.

Surfaces from Scattered Points

25032	*Summary:* Modernization of Surfaces from Scattered Points Sample. The following improvements have been introduced in Surfaces from Scattered Points sample: ▪ Examples for hole filling and data file for Gap filling have been added; ▪ User's Guide has been provided in the documentation; ▪ Installation procedure has been implemented; ▪ The building scripts have been synchronized with other samples; ▪ Support of the obsolete `CSFDB` format has been removed; ▪ The about dialog has been homogenized with other samples; ▪ The problems with incorrect morphing actions, deleting and zooming have been eliminated.

Mesh Framework

24782	*Summary:* Qt based OMF sample. A Qt-based OMF sample has been implemented.
25030	*Summary:* The result of `cut` operation for initial mesh containing isolated objects is not valid. The method `OMFBool_BooleanOperation::CollectMesh` has been modified to take into account elements belonging to isolated sub-meshes. Previously such objects were deleted from the result.
25035	*Summary:* Impossible to collect history during OMF Boolean operations. The flag `CollectHistory`, which allows tracking history of operations, has been introduced in `OMFTest_ModelCommands`. By default it is set to false.
25046	*Summary:* Operation `OMFmeshcut` is not performed when cutting tool has a small size of mesh elements. The check of mesh element area during classification has been corrected in classes `OMFBool_ClassifyPoint` and `OMFBool_SplitElement`.
25131	*Summary:* OMF visualization improvement. New visual mode “Color mesh based on normal vectors direction” is now available in the Mesh Framework Sample (OMF). Express Mesh component now supports storing normal vectors of nodes.
25155	*Summary:* New tool to check validity of meshes for OMF Boolean operations. A new tool, which allows checking if meshes are valid for OMF Boolean operations, has been implemented and available via Draw command `smds_checkboparg`. The following characteristics are taken into account: ▪ links with duplicate nodes; ▪ links and nodes with non-manifold or incoherent mesh; ▪ degenerated edges and triangles; ▪ self-intersection of triangle and edge, triangle and node, edges, edge and node or nodes.
25189	*Summary:* Implement CAM mesh adaptor for OMF. CAM mesh adaptor has been implemented for OMF mesh in package `OMFCAM` to build the shortest path between two points. `OMFCAM` command has been added to test the geodesic algorithm on mesh. Conversion of a point path to a TopoDS_Wire wire is now provided by a separate static method `Geodesic_ShellDistances::BuildPathWirefor`.

Collision Detection

25006	*Summary:* Header `ColDetectionProxy_Intersector.hxx` is confusing The headers in `ColDetection` package have been corrected to include a single guard macro.

DXF Import / Export

23611	*Summary:* Text entity location is wrong reading the attached DXF file. Multiple `if` clauses have been replaced by `switch` in the method `DxfData_TranslateText::MakeShape`.
25033 25375	*Summary:* Enable translation of `HATCH` entities. Support of reading and writing `HATCH` objects has been provided in new classes `DxfFile_RWHatch` and `DxfFile_RWHatchBoundaryEdge`.
25139	*Summary:* Missing faces after import. SAT Reader `DxfFile_FileReader::ReadSatData` has been fixed to provide correct import.
25323	*Summary:* Support Unicode in file names for DXF import/export. DXF import/export procedures now support UTF-8 encoding in input file names.
25325	*Summary:* Invalid import from DXF file (wrong transformation of some entities). The function `DxfData_TranslateInsert::ComputeTrsf` has been improved to provide correct transformation.
25326	*Summary:* Improve robustness of DXF import. The following improvements have been introduced in DXF import product: ▪ Memory usage/performance has been improved in `DxfData_TranslateInsert::MakeShape` for the case when shape transformation does not contain scaling. ▪ Creation of duplicate objects in `DxfData_TranslatePolyline::MakeShape` in case of a 2D or 3D polyline is now avoided. ▪ Translation of polylines degenerated into points has been added in `DxfData_TranslatePolyline::MakeGeometry`. ▪ `DxfData_TranslateDimension::MakeShape` has been protected against accessing null handle. ▪ The call `aDxfModel->AssignHandles()` has been added in method `DxfFile_WorkLibrary::ReadFile` to provide model consistency.

ACIS Import / Export

24962	*Summary:* Crash and warnings importing a file. ACIS translation procedure has been updated to match versions 21500 and later.
25025	*Summary:* Torus-like shape written to SAT or DXF is rendered incorrectly in a 3rd-party software. The method `AcisData_CasCadeToAcis::Wire` has been updated to reorder wires in 2D for U or V periodic case and to avoid exception on empty wires.
25273	*Summary:* Exporting a solid shape made on revolved circle produces bad SAT file. The method `AcisData_CasCadeToAcis::RectangularTrimmedSurface` has been modified to swap U and V parameters for rotation surfaces.
25278	*Summary:* New compilation warnings. The list of parameters has been fixed in functions `NbIntervals` and `Intervals` from `AcisData_CurveLawAdaptor`.

Product Samples

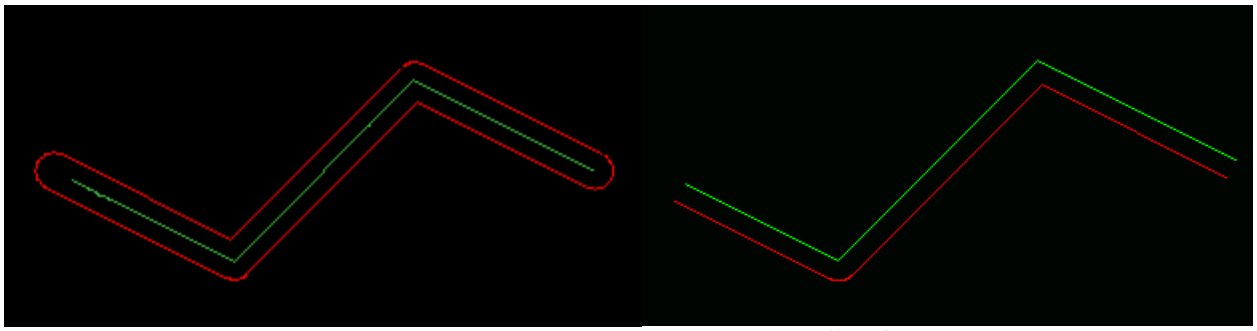
24779 24935 24951 24977 25119	*Summary:* Compilation problems with samples. Product samples have been revised to avoid compilation errors.

New features

New Options for 2D Offset Algorithm

It has become possible to apply offset operation to an open wire, which results now in an open contour.

Previously the offset could be only a closed contour.

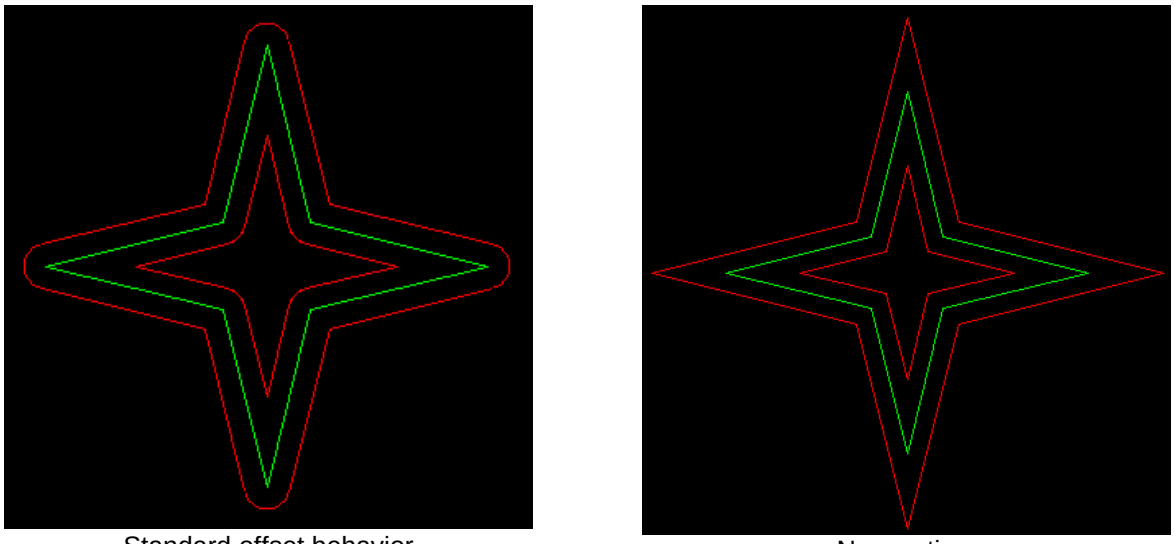


Offset by a closed contour

Offset by an open contour

This feature is provided by the flag `IsOpenResult` implemented in classes `BRepFill_OffsetWire` and `BRepOffsetAPI_MakeOffset`.

Offset on sharp corners can now be created by prolongation until intersection of adjacent edges; previously only arcs of circle could be used.



Standard offset behavior

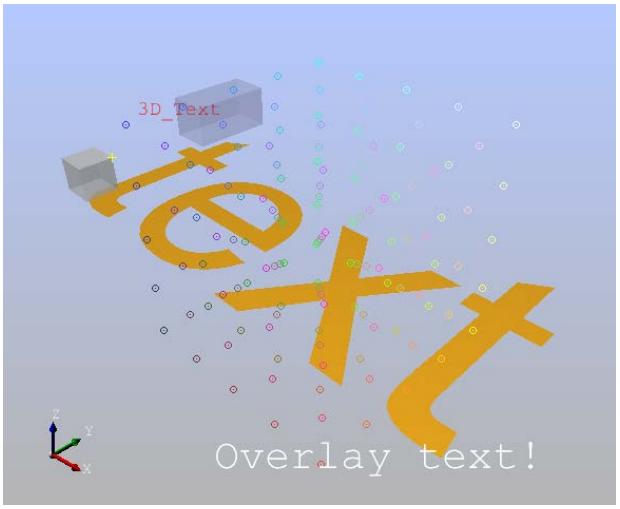
New option.

The behavior depends on the parameter `JoinType` from class `BRepOffsetAPI_MakeOffset.` It can be either `GeomAbs_Arc` or `GeomAbs_Intersection`.

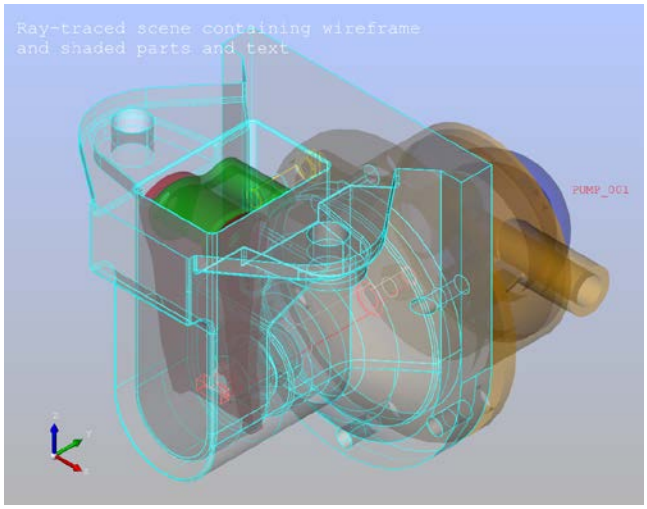
Improvements in Ray Tracing

OCCT ray tracing algorithm has been rewritten using GLSL, which has improved the portability, boosted the performance and enabled the visualization of lines and text in a ray-traced scene with help of OpenGL rasterization (they could not be visualized using purely the ray tracing algorithm). The following benefits also result from this improvement:

- Interactive selection now works in a ray-traced scene exactly in the same way as in a rasterized scene;
- CAD models can be displayed in a ray-traced scene both in shading and wireframe modes;
- Auxiliary scene elements, such as markers and text labels no longer disappear from a 3D view after switching to ray tracing rendering;
- Text annotations can now be added to a ray-traced scene in a common way - i.e. using the standard 2D over-layer of a `V3d_View` instance.

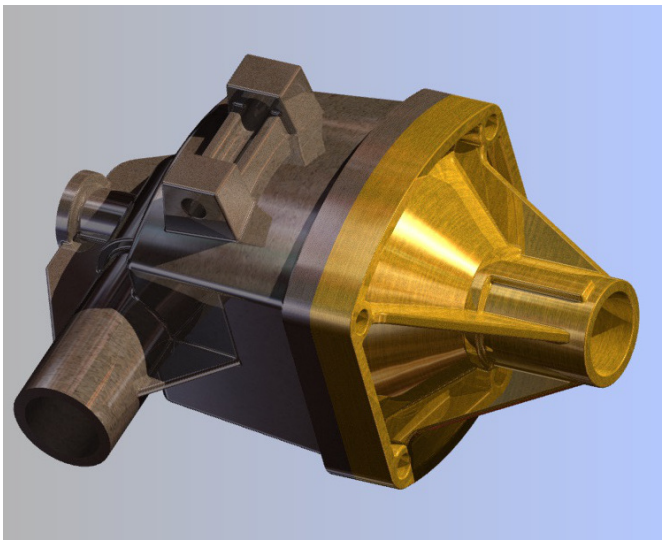


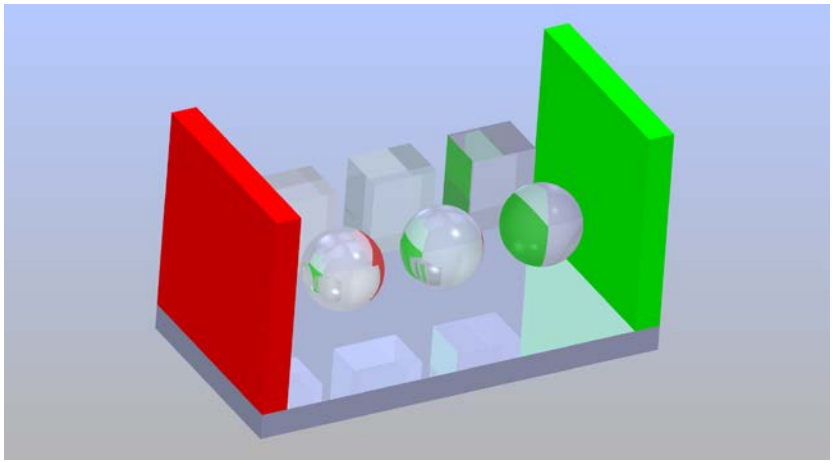
Text in ray-tracing mode



Pump model visualized partly in shading and partly in wireframe mode.

Additionally, texture mapping is now supported in ray tracing mode.

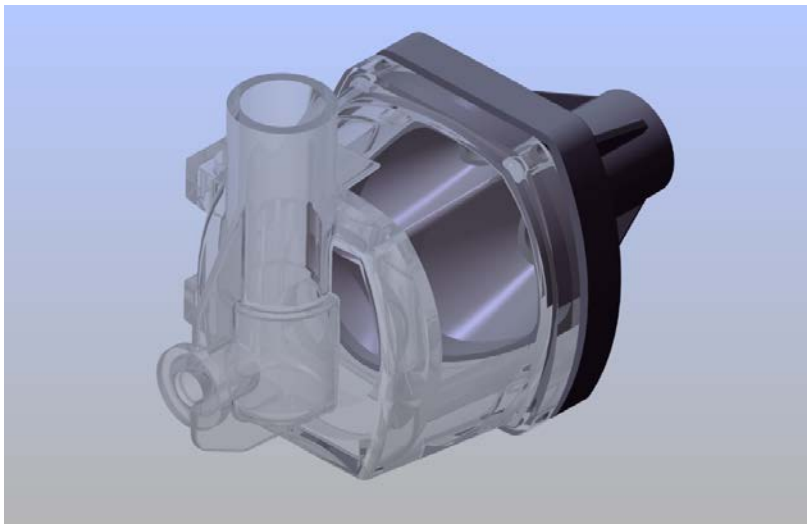




Optical refractions according to the specified refraction index have been implemented for transparent objects.

Refraction index has been added to the attributes of `Graphic3d_MaterialAspect` class. It is taken into account by the ray tracing algorithm when the transparency is non-zero. Three predefined transparent materials have been added to the list of standard materials: glass, diamond and water.

Simulation of a glass cover



Frustum Culling

CPU-based frustum culling is a technique for speeding-up visualization performance on complex scenes, when a significant part of the scene comes out of the camera.

Large CAD models may contain a significant amount of shapes, which become completely invisible for the user with particular camera settings. However, their rendering heavily impacts on the performance (both CPU and GPU). Thus, the main purpose of frustum culling is to filter out the objects located outside of the viewing frustum and skip them during rendering procedure without any visual effect on the resulting image.

The new feature is enabled by default, but can be disabled using `V3d_View::SetFrustumCulling()` method. Note that proper presentation bounding box is essential for this functionality.

The algorithm uses BVH acceleration structures and fits well for static objects. It is possible to specify dynamic objects with mutable flag to optimize scene updates (using `AIS_InteractiveObject::SetMutable()` method).

Improvement of Back Face Culling, Clipping and Capping

Back-face culling reduces the rendered number of triangles (which improves the performance) and eliminates artifacts at shape boundaries. Automatic back-face culling mechanism is controlled by `V3d_View::SetBackFacingModel()` (turned on by default) and is enabled for a specific presentation using `Graphic3d_AspectFillArea3d::SuppresBackfaceFace()` method.

However, this option can be used only for solid objects, where the interior is actually invisible from any point of view, and is known to produce incorrect visual results otherwise.

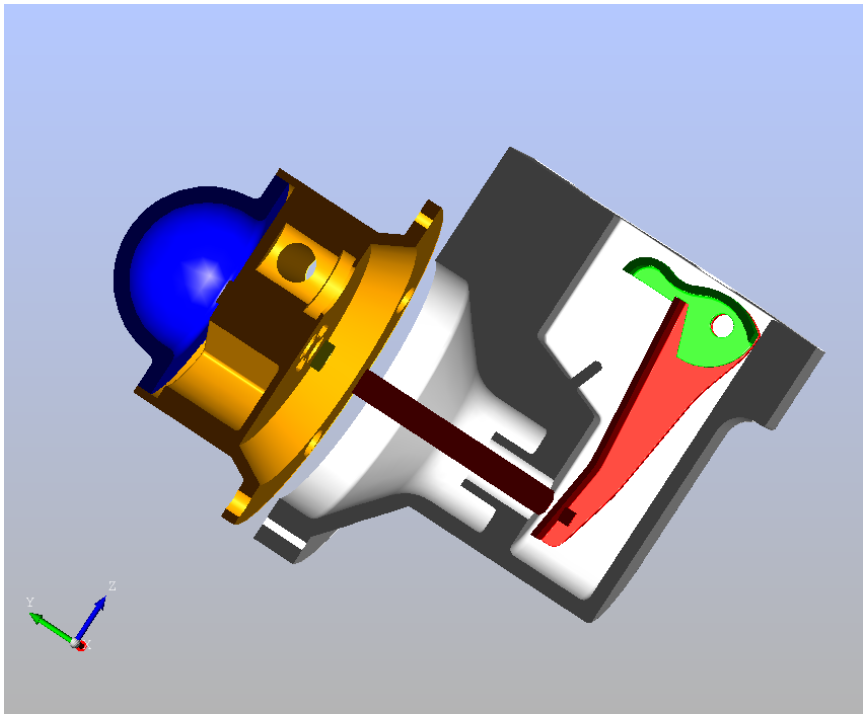
The following problems can arise when the user tries to apply OpenGL clipping planes or to enable capping to visualize a cross-section of a shape:

- Clipping shows the internals of the shape thus back-face culling should be disabled;
- The capping algorithm works incorrectly for non-closed shapes;
- Enabled capping for a closed non-solid shape (e.g. for a closed shell) gives a false impression that the shape contains some material inside.

In case of a complex CAD model, which is typically represented as a compound containing a number of closed and non-closed sub-shapes of various types, the analysis becomes complicated, however, without it back face culling and capping produce incorrect visual results.

Now this problem has been resolved at the level of `StdPrs_ShadedShape` class that traverses sub-shapes and applies the above-mentioned analysis, enabling or disabling back-face culling and capping on sub-shape level fully transparently for application developers.

In the image, there is a complex CAD model of a pump clipped by several planes, with enabled back-face culling and capping



3D View Camera

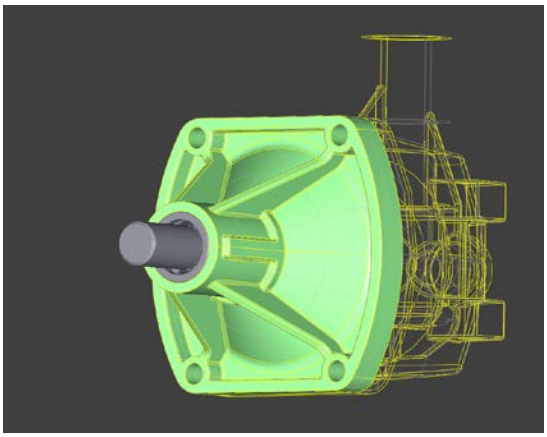
The 3D View Camera is capable to perform correct orthographic and perspective projection in mono or stereo mode.

For backward compatibility, the orthographic mode of camera implementation is close to the old orthographic mode (based on `ViewMapping` and `ViewOrientation`) and inherits its coordinate system (mainly it affects `ZNear` and `ZFar` values, which are counted relatively to the camera center in orthographic mode). Otherwise, perspective mode works as expected with center in `Eye` point.

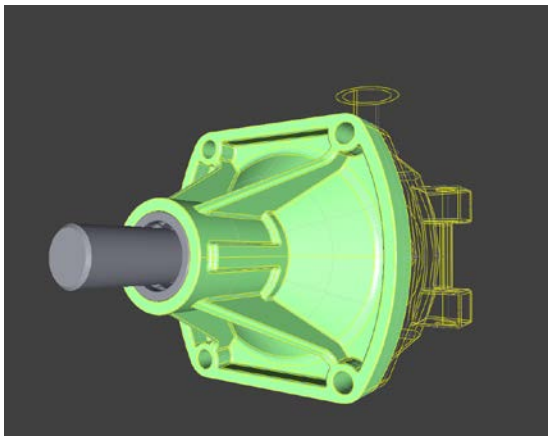
`ZFit` functionality is now automatically performed on scene change (this feature can be disabled), using hardware depth buffer more efficiently. It needs to pass an actual scene bounding box to the camera (now the bounding box for the camera is automatically updated in `Visual3d_View`). Note that proper presentation bounding box is essential for this functionality.

All `V3d` functionality has been ported to the camera. `ViewMapping` and `ViewOrientation` have been completely removed from the OCCT. So, now it is possible to control view either through traditional `V3d_View` API or by accessing `Graphic3d_Camera` object directly.

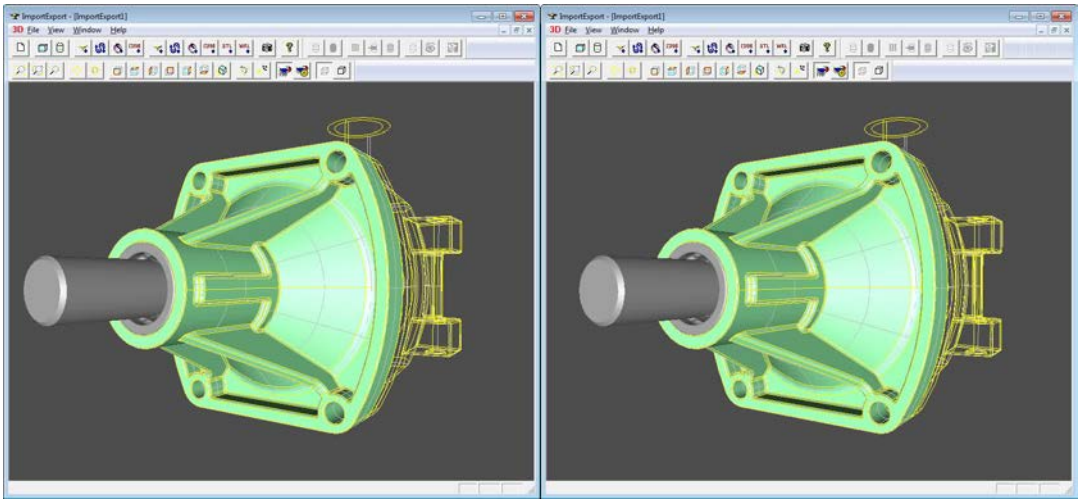
The projection model now can be changed without recreation of `V3d_View` simply by changing the corresponding `V3d_View` property (or by accessing the camera property directly). For this reason classes `V3d_PerspectiveView` and `V3d_OrthographicView` have been removed.



Ball-bearing housing: orthographic camera



Ball-bearing housing: perspective camera



Ball-bearing housing: stereo camera (separate left and right frames)

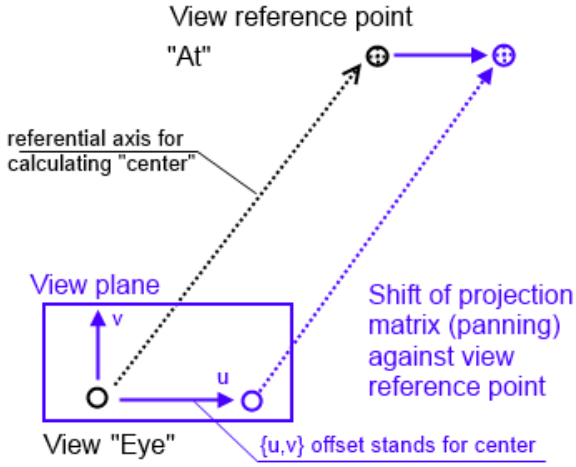
Redesign of viewing transformations based on a camera has also improved environment mapping results as shown of the snapshots below:

In the old mode the reference point was almost always in the center position of the current object, which led to incorrect environment mapping.

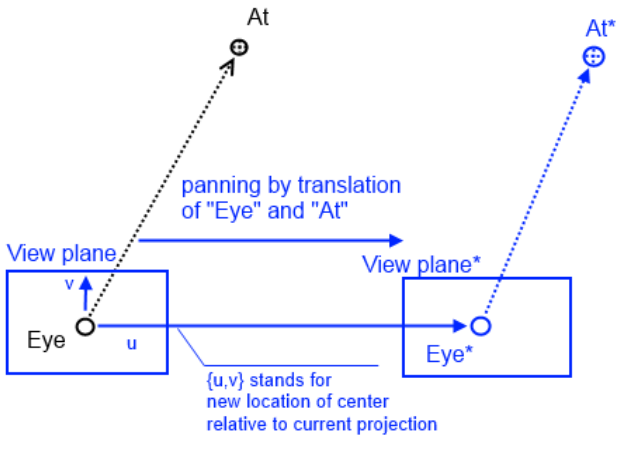
Now in the orthographic camera mode the reference point is propagated on the border of the bounding box (and in the perspective mode it also moves with the camera).

Elimination of Projection Shift Concept

The projection shift concept has been removed. The reference system of the view is now located at the center of screen, in consistence with definition rules of a camera. There is no more `Center` property of `V3d_View`, which was originally used to specify shift of projection matrix and to slide reference system of the view along the screen.



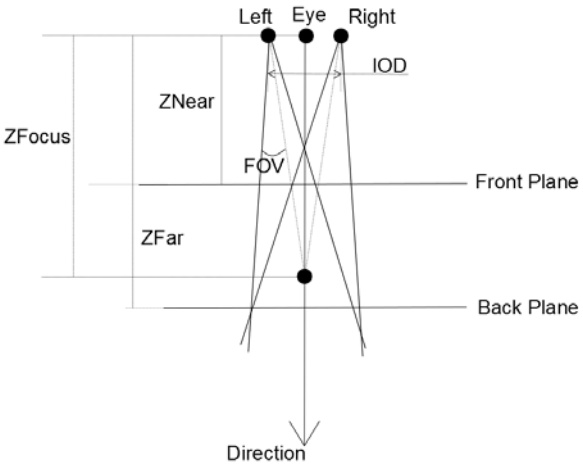
Old implementation



New implementation

Stereographic view

3D viewer now supports stereoscopic output on systems providing OpenGL Quad Buffer (professional graphics). Intraocular distance and Z focus values can be adjusted through the new camera interface in `V3d_View`.
In stereographic projection mode the camera prepares two projection matrices to display different stereo-pictures for the left and for the right eye. The principle of stereographic view can be seen in the scheme:

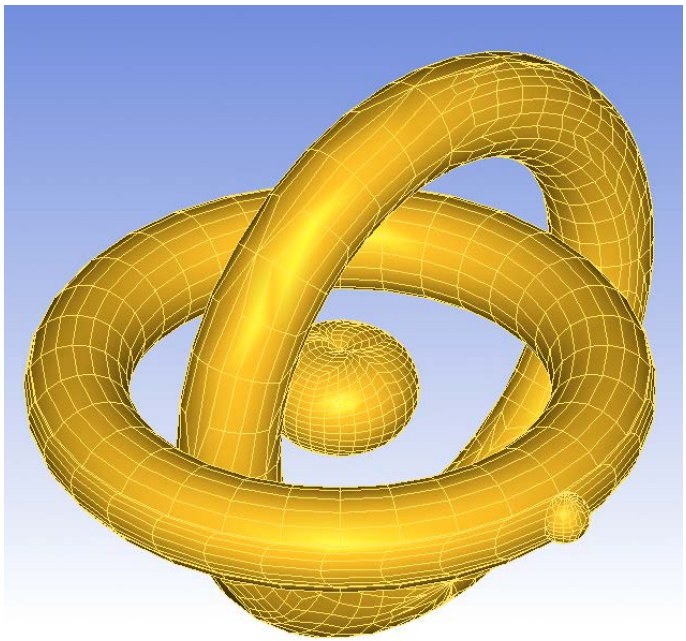


IOD (intraocular distance) can be defined by the absolute value or relatively to the focal length of the the camera.
FOV (field of view) defines the field of camera view by y axis in degrees (45° is default).
ZFocus – defines the distance to the point of stereographic focus.

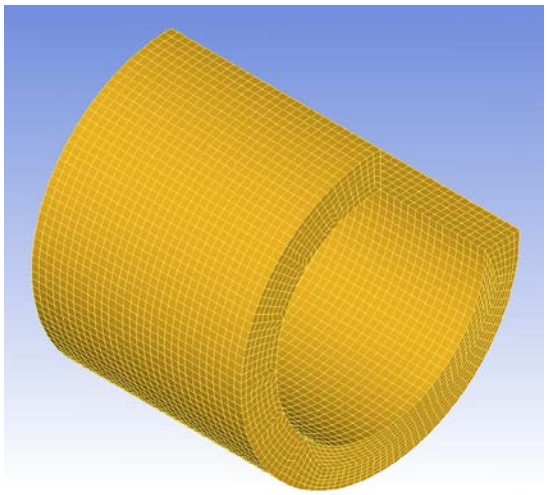
To enable stereo projection, your workstation should meet the following requirements:

- the graphic card should support quad buffering;
- you need active 3D glasses (LCD shutter glasses);
- The graphic driver needs to be configured to impose quad buffering for newly created OpenGl contexts; the viewer and the view should be created after that.

Express Mesh: Quad-only mesh



The option to produce a quad-only mesh has been introduced in Express Mesh product. The mesh is generated by the algorithm of recursive contour subdivision with pattern matching.



Porting to version 6.8.0

Porting of user applications from the previous 6.7.1 OCCT version to version 6.8.0 requires the following issues to be taken into account:

Changes in NCollection classes

Method `Assign()` in `NCollection` classes does not allow any more copying between different collection types. Such copying should be done manually.

List and map classes in `NCollection` package now require that their items be copy-constructible, but do not require items to have default constructor. Thus the code using `NCollection` classes for non-copy-constructible objects needs be updated. One option is to provide copy constructor; another possibility is to use Handle or other smart pointer.

3D View Camera

If `ViewMapping` and `ViewOrientation` were used directly, this functionality has to be ported to the new camera model. The following methods should be considered as an alternative to the obsolete `Visual3d` services (all points and directions are supposed to be in world coordinates):

- `Graphic3d_Camera::ViewDimensions()` or `V3d_View::Size()/ZSize()` - returns view width, height and depth (or "Z size"). Since the view is symmetric now, you can easily compute top, bottom, left and right limits. `Graphic3d_Camera::ZNear()/ZFar()` can be used to obtain the near and far clipping distances with respect to the eye.
- `Graphic3d_Camera::Up()` or `V3d_View::Up()` - returns Y direction of the view.
- `Graphic3d_Camera::Direction()` returns the reverse view normal directed from the eye, `V3d_View::Proj()` returns the old-style view normal.
- `Graphic3d_Camera::Eye()` or `V3d_View::Eye()` - returns the camera position (same as projection reference point in old implementation).
- `Graphic3d_Camera::Center()` or `V3d_View::At()` - returns the point the camera looks at (or view reference point according to old terminology).

The current perspective model is not fully backward compatible, so the old perspective-related functionality needs to be reviewed.

Please revise application-specific custom presentations to provide proper bounding box. Otherwise object might become erroneously clipped by automatic `ZFit` or frustum culling algorithms enabled by default.

Redesign of Connected Interactive Objects

The new implementation of connected Interactive Objects makes it necessary to take the following steps if you use connected Interactive Objects in your application.

- Use new `PrsMgr_PresentableObject` transformation API.
- Call `RemoveChild()` from the original object after connect if you need the original object and `AIS_ConnectedInteractive` to move independently.
- Access instances of objects connected to `AIS_MultiplyConnectedInteractive` with `Children()` method.
- For `PrsMgr_PresentableObject` transformation:
  - `SetLocation (TopLoc_Location) -> SetLocalTransformation (gp_Trsf);`
  - `Location -> LocalTransformation`
  - `HasLocation -> HasTransformation`
  - `ResetLocation -> ResetTransformation`

Support of UNICODE Characters

Support of UNICODE characters introduced in OCCT breaks backward compatibility with applications, which currently use filenames in extended ASCII encoding bound to the current locale. Such applications should be updated to convert such strings to UTF-8 format.
The conversion from UTF-8 to `wchar_t` is made using little-endian approach. Thus, this code will not work correctly on big-endian platforms. It is needed to complete this in the way similar as it is done for binary persistence (see the macro `DO_INVERSE` in `FSD_FileHeader.hxx`).

Elimination of Projection Shift Concept

It might be necessary to revise the application code, which deals with `Center()` method of `V3d_View`.

This method was used to pan a `V3d` view by virtually moving the screen center with respect to the projection ray passed through `Eye` and `At` points. There is no more need to derive the panning from the `Center` parameter to get a camera-like eye position and look at the coordinates. `Eye()` and `At()` now return these coordinates directly. When porting code dealing with `Center()`, the parameters `Eye()` and `At()` can be adjusted instead. Also `V3d_View::SetCenter(Xpix, Ypix)` method can be used instead of `V3d_View::Center(X, Y)` to center the view at the given point. However, if the center coordinates X and Y come from older OCCT releases, calling `V3d_View::Panning(-X, -Y)` can be recommended to compensate missing projection shift effect.

There are several changes introduced to `Graphic3d_Camera`. The internal data structure of the camera is based on `Standard_Real` data types to avoid redundant application-level conversions and precision errors. The transformation matrices now can be evaluated both for `Standard_Real` and `Standard_ShortReal` value types. `ZNear` and `ZFar` planes can be either negative or positive for orthographic camera projection, providing a trade-off between the camera distance and the range of ZNear or ZFar to reduce difference of exponents of values composing the orientation matrix - to avoid calculation errors. The negative values can be specified to avoid Z-clipping if the reference system of camera goes inside of the model when decreasing camera distance.

The auto z fit mode, since now, has a parameter defining Z-range margin (the one which is usually passed as argument to `ZFitAll()` method). The methods `V3d_View SetAutoZFitMode(), AutoZFitScaleFactor(), ZFitAll()` deals with the new parameter.

The class `Select3D_Projector` now supports both orientation and projection transformation matrices, which can be naturally set for the projector. The definition of projector was revised in `StdSelect_ViewerSelector3d`: perspective and orthographic projection parameters are handled properly. Orthographic projector is based only on direction of projection - no more Center property. This makes it possible to avoid unnecessary re-projection of sensitive while panning, zooming or moving along the projection ray of the view. These operations do not affect the orthographic projection.

Appendix

The following unused packages have been removed:

```
PColPGeom,
PcolPGeom2d
```

All classes have been removed from package `PCollection` except for:

```
PCollection_HArray,
PCollection_HAsciiString,
PCollection_HExtendedString,
PCollection_HSequence,
PCollection_SeqExplorer,
PCollection_SeqNode,
```

The following unused classes have been removed:

```
AppBlend_Line,
AppBlend_SectionGenerator,
AppCont_SurfLeastSquare,
AppCont_TheLineTool,
AppCont_TheSurfTool,
AppParCurves_MLineTool,
AppParCurves_Projection,
ApproxInt_WLine,
Approx_ComputeCSurface,
Approx_TheLineTool,
Blend_Iterator,
Contap_ArcTool,
Contap_SurfaceTool,
Contap_TopolTool,
Dynamic_EnumerationParameter,
Dynamic_MethodInstance,
Extrema_ExtPSOfRev,
Geom_Array1OfBoundedCurve,
Geom_(H)Array1OfBoundedSurface,
Geom_Harray1OfBoundedCurve,
Geom_(H)Array2OfBoundedSurface,
Geom_(H)Array2OfBSplineSurface,
Geom_HArray2OfBezierSurface,
Geom_(H)SequenceOfBoundedSurface,
Geom_HSequenceOfSurface,
Geom_(H)Array1OfGeometry,
Geom_(H)Array1OfBoundedCurve,
Geom_HSequenceOfGeometry.
GProp_CurveTool
GProp_DomainTool
GProp_FaceTool.
GccGeo_ParGenCurve,
IntCurveSurface_SurfaceTool,
Intf_InterferencePolygon3d,
NCollection_(H)Set,
NCollection_Define(H)Set,
NCollection_Slist,
Ncollection_DefineSList,
TCollection_(H)Set,
TCollection_SetIterator,
TCollection_SetList,
Tcollection_Queue,
TCollection_QueueNode,
TCollection_AVLBaseNode,
```

TCollection_AVLIterator,
TCollection_AVLList,
TCollection_AVLNode,
TCollection_AVLSearchTree,
TCollection_AVLBaseNode,
TCollection_AVLIterator,
TCollection_AVLList,
TCollection_AVLNode,
TCollection_AVLSearchTree,
TCollection_Stack,
TCollection_StackIterator,
TCollection_StackNode.

The following duplicate and unused instantiations of TCollection classes have been removed:

BRepMAT2d_SequenceOfBasicElt (duplicates MAT_SequenceOfBasicElt),
BRepMesh_NodeHasherOfDataStructureOfDelaun,
BRepMesh_IDMapOfNodeOfDataStructureOfDelaun,
QADNaming_DataMapOfShapeOfName (duplicates Dnaming_DataMapOfShapeOfName),
TDF_LabelLabelMap,
Graphic3d_SequenceOfAddress,
CDM_StackOfDocument (duplicates ListOfDocument),
Draw_MapOfFunctions (duplicates Plugin_MapOfFunctions),
ExprIntrp_SkackOfNames (duplicates TcolStd_ListOfAsciiString),
(H)SeqOfPersistent, MapPSDHasher and (H)SeqOfCallback from package Storage.

The following instantiations of generic classes have been converted to non-generic:

AppDef_LinearCriteria,
AppDef_SmoothCriterion,
AppDef_Variational,
BRepApprox_ApproxLineGen,
BRepBlend_HCurve3dToolGen,
BRepBlend_HCurve2dToolGen.
BRepApprox_SurfaceToolGen,
Contap_ArcFunction,
Contap_ContourGen,
Contap_HContToolGen,
Contap_HCurve2dToolGen,
Contap_Line,
Contap_Point,
Contap_SurfFunction,
Contap_SurfProps,
IntWalk_PWalking,
math_DoubleTab,
math_SingleTab,
XmlObjMgt_Array1,

The following generic classes have been converted to non-generic and renamed:

Intf_InterferencePolyhedron to IntPatch_InterferencePolyhedron;
MoniTool_Elem to MoniTool_TransientElem;
Transfer_SimpleBinder to TransferBRep_BinderOfShape;
Geom2dInt_CurveTool to Geom2dInt_Geom2dCurveTool;
MAT2d_BisectingLocus to BRepMAT2d_BisectingLocus;
MAT_Mat to MAT2d_Mat2d;
GccGeo_Circ2dTanOn to Geom2dGcc_Circ2dTanOnGeo;
GccGeo_Circ2d2TanRad to Geom2dGcc_Circ2d2TanRadGeo;
GccGeo_Circ2d2TanCen, to Geom2dGcc_Circ2d2TanCenGeo;
GccGeo_Circ2d2TanOnRad, to Geom2dGcc_Circ2d2TanOnRadGeo;
GccGeo_CurvePGTool to Geom2dGcc_CurveToolGeo;
IntImp_ZerCOnSSParFunc to IntPatch_CSFunction;

LProp_FuncCurExt to Geom2dLProp_FuncCurExt;
LProp_FuncCurNul to Geom2dLProp_FuncCurNul;
LProp_NumericCurInf to Geom2dLProp_NumericCurInf;
GccEnt_QualifiedCurv to Geom2dGcc_QCurve;
HatchGen_ElementGen to Geom2dHatch_Element;
HatchGen_ElementsGen to Geom2dHatch_Elements;
HatchGen_HatchingGen to Geom2dHatch_Hatching;
HatchGen_Hatcher to Geom2dHatch_Hatcher;
Extrema_GExtCC to Extrema_ExtCC;
Extrema_GExtCC2d to Extrema_ExtCC2d;
Extrema_GLocateExtCC to Extrema_LocateExtCC;
Extrema_GLocateExtCC2d to Extrema_LocateExtCC2d;
GccIter_Circ2d2TanOn to Geom2dGcc_Circ2d2TanOnIter;
GccIter_Circ2d3Tan to Geom2dGcc_Circ2d3TanIter;
GccIter_Lin2d2Tan to Geom2dGcc_Lin2d2TanIter;
GccIter_Lin2dTanObl to Geom2dGcc_Lin2dTanOblIter;
GProp_CGProps to BRepGProp_Cinert;
GProp_SGProps to BRepGProp_Sinert;
GProp_VGProps to BRepGProp_Vinert;
GProp_VGPropsGK to BRepGProp_VinertGK;
Blend_Extremity to BRepBlend_Extremity;
Blend_Line to BRepBlend_Line;
Blend_PointOnRst to BRepBlend_PointOnRst.

Supported Platforms and Pre-requisites

Open CASCADE Technology is supported on Windows (IA-32 and x86-64), Linux (x86-64) and MAC OS X (x86-64) 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.org/getocc/require/.

Linux Operating System	Mandriva 2010, CentOS 5.5, CentOS 6.3, Fedora 17, Fedora 18, Ubuntu-1304, Debian 6.0*
Windows Operating System	MS Windows 8 / 7 SP1 / Vista SP2 / XP SP3
Mac OS X Operating System	Mac OS X 10.9 Mavericks / 10.8 Mountain Lion / 10.7 Lion / 10.6.8 Snow Leopard
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 1.1+ (OpenGL 3.3+ is recommended)
C++ *For Linux:* *For Windows:* *For Mac OS X:*	 GNU gcc 4.0. - 4.7.3. Microsoft Visual Studio 2005 SP1 with all security updates Microsoft Visual Studio 2008 SP1 Microsoft Visual Studio 2010 SP1** Microsoft Visual Studio 2012 Update 4 Microsoft Visual Studio 2013 Update 2 Intel C++ Composer XE 2013 SP1 XCode 3.2 or newer (4.x is recommended)
TCL (for testing tools) *For Linux:* *For Windows:* *For OS X:*	 Tcltk 8.5 or 8.6 http://www.tcl.tk/software/tcltk/8.6.html ActiveTcl 8.5 or 8.6 http://www.activestate.com/activetcl/downloads Built-in Tcl/Tk 8.5
Qt (for demonstration tools)	Qt 4.8.6 http://qt-project.org/downloads
FreeType (OCCT Text rendering)	FreeType 2.4.11-2.5.3 http://sourceforge.net/projects/freetype/files/
FreeImage (Support of common graphic formats)	FreeImage 3.16.0 http://sourceforge.net/projects/freeimage/files/Source%20Distribution/
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 3.x or 4.x http://www.threadingbuildingblocks.org/
Doxygen (optional for building documentation)	Doxygen 1.8.5 http://www.stack.nl/~dimitri/doxygen/download.html

- * Debian 60 64 bit is a permanently tested platform.
- ** The official release of OCCT for Windows contains libraries built with VC++ 2010.