Open CASCADE Technology and Products ver. 6.9.0

Release Notes

Overview

Open CASCADE Technology and Products version 6.9.0 is a minor release, which includes about 400 improvements and bug fixes over the previous release 6.8.0.

Version 6.9.0 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 6.9.0.

Highlights

Modeling algorithms:

- **Fuzzy Boolean operations - possibility to specify global tolerance for a Boolean operation;**
- **Support of multiple arguments for a single Boolean operation;**
- **Improved performance and thread safety;**
- **Shape proximity detector;**
- **Revised usage of `Closed` flag in shapes;**
- **Precise evaluation of edge tolerance;**
- **Additional options to tune `BRepMesh` algorithm;**
- **More robust algorithms of surface-surface intersection and curve on surface projection;**
- **Improved support of user feedback messages in Shape Processing;**
- **New tool to eliminate small solids in `ShapeFix`;**

Visualization:

- **Redesign of selection mechanism for better performance;**
- **OpenGL ES 2.0 compatibility improvements;**
- **Support of OpenGL viewer on iOS and ray tracing on OS X;**
- **Robust implementation of immediate mode using FBO;**
- **Option to disable automatic re-triangulation of shapes on display;**

Data Exchange:

- **Reconstruction of p-curves optimized to accelerate import from STEP;**
- **Colors of edges and faces are written to IGES 5.3;**
- **Orientation of faces is preserved on export to IGES 5.1;**
- **Export to STL and VRML 2.0 corrected;**

Other

- **Support of parallelism without TBB library;**
- **64-bit mode becomes default on Windows;**
- **Improved stability of performance measurements in tests;**

Products:

- **Express Mesh : improved quality of mesh near face boundary;**
- **C# and Java wrappers: Support of SWIG 3.x;**
- **DXF Import: Reading of the recent DXF versions with ACIS data encoded in binary form;**
- **ACIS SAT Import: Reading of SAB (Standard ACIS Binary) files.**

Table of Contents

Modifications

Foundation Classes

23626	*Summary:* Impossible to open files containing localization characters in the name. `Draw_VariableCommands.cxx` has been corrected to allow opening and saving files containing localization characters in the name.
24826	*Summary:* Wrapping of parallelization algorithms. Simple primitives have been implemented to parallelize loops of types `for` and `foreach`. The primitives encapsulate complete logic for creating and managing parallel context of loops. They can also serve as wrapper for TBB library primitives. To use them, is necessary to implement a TBB like interface based on functors. If a parallelized loop iterates on the collections with direct access by index (such as `Vector` or `Array`), it is more efficient to use the primitive `ParallelFor` because it has no critical section.
24285	*Summary:* Updates of `PLib::EvalPolynomial` for code acceleration. The functions `PLib::EvalPolynomial` and PLib::NoDerivativeEvalPolynomial have been refactored to work faster: ▪ Iteration by degree is made in the outer loop; ▪ Pointer arithmetic is avoided; ▪ Recursive templates are used to expand loop by dimension in specific cases.
25514	*Summary:* `TKernel, OSD_Timer` - do not accumulate error in timer within queries in running state. The methods `OSD_Timer::ElapsedTime(), OSD_Timer::Show()` and `OSD_Chronometer::Show()` have been fixed to avoid accumulation of error due to queries in running state.
25559	*Summary:* `SIGSEGV` in `TKMath` when computing max tolerance of curve on surface. `BOPTools_CheckCurveOnSurface` class has been corrected to avoid trying to compute a function out of domain of definition. `math_Recipes::LU_Decompose` now works with arguments `NaN, Inf` and `Ind`.
25608	*Summary:* `TKernel, NCollection_UtfIterator` - fix iteration of surrogate pairs in UTF-16. Iteration of surrogate pairs in UTF-16 has been fixed in `NCollection_UtfIterator`.
25630	*Summary:* Possible memory leaks in `BRepGProp_Vinert` and `BRepGProp_Sinert.` The classes `RepGProp_Sinert` and `BRepGProp_Vinert` have been refactored: ▪ All static variables have been removed. ▪ Common functionality connected with Gauss integration has been moved to the new `BRepGProp_Gauss` class.

25717	*Summary:* Non reentrant (and hence non-thread-safe) `math_RealRandom / _IntegerRandom`. Classes `math_IntegerRandom` and `math_RealRandom` and method `math_Recipes::Random2` have been removed. Class `math_BullardGenerator` is used instead.
25963	*Summary:* Define `HashCode()` for an unsigned `int` within `armv7` target. `OSD_EnvironmentIterator` has been modified to avoid using `_NSGetEnviron()` on iOS.

Modeling Data

24411	*Summary:* `SplitShape` produces shape with incorrectly parameterized periodic 3D curve. Additional flag `theAdjustPeriodic` has been added in constructors and methods `SetTrim()` of classes `Geom_TrimmedCurve` and `Geom2d_TrimmedCurve` to indicate whether adjustment of the range on the periodic curve to be inside the first period is necessary.
25479	*Summary:* `BRepTools::Clean()` cleans all edge polygons, even related to different shapes. The method `BRepTools::Clean()` has been modified to iterate on edges and clean associated polygons only on triangulations related to faces composing the given shape.
25489	*Summary:* `GCPnts_TangentialDeflection` gives incorrect distribution of points. Handling of the last point has been improved in method `GCPnts_TangentialDeflection::PerformCurve()`. Now the interval is checked with this point instead of adding it to the output set of points.
25545	*Summary:* `TopLoc_Location::Transformation()` provokes data races. The class `TopLoc_Location` has become thread-safe. Unused methods have been removed.
25649	*Summary:* The command `crvtpoints` return wrong deflection in output. The algorithm PSO + Newton Minimum has been implemented for deflection computation in `GeometryTest_CurveCommands`.
25656 26027	*Summary:* Specification of semantic of Closed flag of an edge. The use of `Closed` flag has been unified throughout OCCT: ▪ This flag is meaningful for `TopoDS_Wire` and `TopoDS_Shell` only, because these entities may hedge an area in 2D space or a volume in 3D space correspondingly; ▪ Other types of `TopoDS` shapes should not have this flag; ▪ The change of this flag is controlled by high-level algorithms (not `BRep_Builder`). The check of closed state of edges is added in method `BRep_Tool::IsClosed()`. An edge is closed if and only if its first and last vertices are the same.

25706	*Summary:* SIGSEGV after making a BSplineCurve rational. The exception arising after conversion of a non-rational B-spline to rational has been eliminated in method Geom2d_BSplineCurve::ValidateCache(). New command setweight has been implemented in DRAW to change weights of B-splines.
25860	*Summary:* Buffer overrun in TopTools_ShapeSet::Read. The method TopTools_ShapeSet::Read has been corrected to avoid reading/writing out of array bounds.
25971	*Summary:* Close B-Spline knots are merged after save/restore or export/import. The minimum precision used to consider B-Spline knots as identical has been increased to Precision::PConfusion() in method Geom2d_BSplineCurve::Segment().
26148	*Summary:* BRep_Tool::IsClosed failed to judge a closed edge on Poly_Triangulation. The argument TopLoc_Location has been added in the signature of method BRep_Tool::IsClosed().

Modeling Algorithms

17129	*Summary:* `ShapeFix` projector makes 2d curves with oscillations. The case of uneven point distribution is now taken into account in class `ShapeConstruct_ProjectCurveOnSurface`.
21727	*Summary:* `BRepBuilderAPI_Copy` (and possibly other similar tools) create new shape in `Frozen` state. The flag `Frozen` is now set to false for top-level shapes created by `BrepTools_Modifier`.
22598	*Summary:* Approximation of p-curve by 2D line. The new method `ShapeConstruct_ProjectCurveOnSurface::getLine()` checks if the projected p-curve can be approximated by a straight line. This is done before full-scale projection, to improve the performance. If it is straight, a pcurve is created as Line only if it will have the same range parameterization as a 3D curve; otherwise a Bspline of degree 1 is created. Re-approximation of linear pcurves by Bsplines has been removed from `ShapeFix_Edge`.
24161	*Summary:* Boolean operation hanging. The walking algorithm has been fixed in method `IntWalk_Pwalking::Perform()` to improve work of Boolean operations with mirror solids.
24643	*Summary:* No section curve between plane and cone. The tolerance of solution point has been increased according to the precision of intersection in method `IntStart_SearchOnBoundaries::PointProcess()`.
24646	*Summary:* Wrong result done by Boolean Operation algorithm. The following improvements have been introduced in Boolean Operations algorithm: ▪ Creation of intersection vertices is now avoided in class `BOPAlgo_PaveFiller::PerformEF()` if the intersection lies `ON` the face boundary. ▪ New method `IntTools_Context::IsPointInFace()` has been implemented to check if the point is IN the face. ▪ Method `IntTools_EdgeEdge::FindBestSolution()` now processes the touching cases.
24697	*Summary:* Exception is raised during projection of the curve on the surface. The algorithm trimming periodic curves has been changed in method `GeomProjLib::Curve2d()`. Now the curve is trimmed in the surface boundaries.
24803	Improve the result of v/v interference for two vertices case. The case of vertex/vertex interference is now taken into account by method `BOPTools_AlgoTools::MakeVertex()`.

24988	*Summary:* Wrong result done by projection algorithm. The projection algorithm `ProjLib_ComputeApprox` has been modified to project the whole curve without extension of the result. Classes `AppCont_Function` and `AppCont_FunctionTool` have been united into class `AppCont_Function`, which also provides information about periodicity.
25124	*Summary:* Removal of continuity checks for offset geometries. It has become possible to create an offset curve/surface with C0-continuity if it actually has a G1-continuity. The checks for curves or surfaces, which are defined as C0 but have G1 continuity, have been added in methods `SetBasisCurve` and `SetBasisSurface` from `Geom_OffsetCurve` class. The flag `isNotCheckC0` has been added to constructors of classes `Geom2d_OffsetCurve`, `Geom_OffsetCurve` and `Geom_OffsetSurface` to disable check of continuity (i.e. an offset can be built from a C0-curve/surface); however an unexpected or even invalid result can be obtained. Use this option carefully.
25193	*Summary:* Bad Intersection curve obtained by Surface/Surface Intersection Algorithm. Condition on exit from loop has been corrected in method `IntWalk_Pwalking::Perform()`. Tolerance of confusion for vertices on curve has been increased in method `IntPatch_Wline::ComputeVertexParameters()` to avoid loops on curve.
25321	*Summary:* Test case hangs in Debug mode on Debian60-64 platform. Infinite loop arising during projection of a curve on surface has been eliminated in method `Approx_ComputeCLine::Perform()`.
25398	*Summary:* Provide shape proximity detector. Shape proximity detector has been implemented in new classes `BrepExtrema_TriangleSet` and `BrepExtrema_ShapeProximity` – see New Features section.
25416 25772 25991	*Summary:* Wrong section curve. The following improvements have been introduced to obtain better section curves: ▪ Restriction line is processed in `IntTools_FaceFace` using methods of `GeomInt_IntSS` class. ▪ It is now checked in `IntPatch_ImpPrmIntersection` class if Restriction- and Walking-lines (or Restriction-Restriction lines) coincide. ▪ It is now checked in `IntPatch_ImpImpIntersection` class if `Rline` and `Gline` coincide. ▪ New class `IntPatch_PointLine` inherited from `IntPatch_Line` has been implemented.
25432	*Summary:* Wrong result obtained by `MakerVolume` operator. The method `BOPTools_AlgoTools::FindPointInFace` now starts binormal calculation from the point located outside of the tolerance circle of the edge.

25449	*Summary:* Excess vertex in result of General Fuse operation. The parametric size of the contact zone between vertex and face has been changed in method `IntTools_EdgeFace::CheckTouchVertex`.
25450	*Summary:* Common operation returns wrong shape. The face index has been added in post-treatment map in method `BOPAlgo_PaveFiller::PerformEF()`.
25451	*Summary:* BRepFilletAPI_MakeFillet fails on customer's shape when a small radius of fillet is given. The confusion tolerance is now defined taking into account the edge tolerance in the local function `isTangentFaces` of `ChFi3d_Builder`.
25453	*Summary:* `SIGSEGV` in `BrepFill_Sweep::BuildShell`. The problem with exception caused by the degenerated first edge in the generated sweep has been fixed in `BrepFill_Sweep`.
25456	*Summary:* `BOPAlgo_CheckerSI` reports an error on the given shape. The value of testing parameter has been changed for cases of infinite curves in method `IntTools_FaceFace::MakeCurve`.
25460	*Summary:* BRepOffsetAPI_ThruSections fails on a wing with 11 sections. The algorithm inserting new knots has been corrected in method `GeomFill_Profiler::Perform` to avoid different processing of confusing knots from section to section.
25465	*Summary:* Excess vertex in the result of `CUT` operation. Methods `ClearVertexes()`, `RemoveVertex(…)` and `InsertVertexBefore(…)` have been added in class `IntPatch_Wline`.
25468	*Summary:* `GeomConvert_ApproxSurface` should have a constructor for adaptors. Constructors for `Adaptor3d_Surface` objects have been implemented in classes `GeomConvert_ApproxCurve`, `GeomConvert_ApproxSurface` and `Geom2dConvert_ApproxCurve`.
25470	*Summary:* Wrong result of `COMMON` operation. The method `BOPAlgo_BuilderFace::PerformAreas()` has been fixed to correct splitting of infinite faces.
25477 25722	*Summary:* Boolean Operations with additional tolerance – Fuzzy Boolean operations. Fuzzy Boolean operations allow performing Boolean operations on the shapes with close coincidence between the entities of these shapes, i.e. between shapes in which some entities from one shape are intended to coincide with some entities from the other, but the coincidence is not precise. See also New Features section.

25477 25722	The optional additional tolerance (Fuzzy Logic) been added to the following classes: `BOPAlgo_ArgumentAnalyzer`, `BOPAlgo_BOP`, `BOPAlgo_Builder`, `BOPAlgo_MakerVolume`, `BOPAlgo_PaveFiller`, `BOPDS_DS`, `BRepAlgoAPI_BooleanOperation`, `BRepAlgoAPI_Check`, `BRepAlgoAPI_Common`, `BRepAlgoAPI_Cut`, `BRepAlgoAPI_Fuse` and `BRepAlgoAPI_Section`. Two new classes `BRepAlgoAPI_Algo` and `BRepAlgoAPI_BuilderAlgo` have been introduced to provide the root interface for algorithms.
25480	*Summary:* Incorrect result of `BRepOffsetAPI_MakePipe`. The algorithm of elimination of inner locations of profiles has been corrected in method `BrepFill_Pipe::Perform`.
25487	*Summary:* `Extrema_GenExtPS` needs to be optimized. Cache usage has been improved in class `Extrema_GenExtPS`.
25488	*Summary:* Wrong result of two trimmed cylinders intersection. New function `IntSurf_PntOn2S::IsSame` returns TRUE if 2D- and 3D-coordinates of `theOuterPoint` are equal to the corresponding coordinates of me (with given tolerance). It is forbidden to insert additional points if the existing Wline contains only two coinciding points.
25491	*Summary:* `BRepOffsetAPI_MakeOffset` algorithm crashes on a shape with a big offset value. Method `BrepFill_OffsetWire::UpdateDetromp` has been modified to provide correct processing of `GeomAbs_Intersection` mode.
25494	*Summary:* Wrong result obtained by projection algorithm. The calculation of the last parameter of curve projected to a surface of revolution, if the curve is intersected with the axis of revolution has been changed in method `ProjLib_ProjectedCurve::Load`.
25504	*Summary:* Exception raised during projection curve on surface. The method `ProjLib_PrjResolve::ProjLib_PrjResolve` has been modified to avoid moving the projected point to the surface boundary.
25505	*Summary:* General Fuse produces self-intersection shape. A misprint has been fixed in method `BOPAlgo_BuilderSolid::PerformAreas()`.
25509	*Summary:* Wrong shape considered as valid by `checkshape`. New class `BrepCheck_Solid` has been implemented. It checks the following features of solids: ▪ Shells that overlap each other; ▪ Detached parts of the solid (vertices, edges) that have non-internal orientation ▪ Shells containing entities of the solid that are outside of the shells ▪ Shells that enclose other Shells

25531	*Summary:* Difference in intersection result on Windows and Linux platform is very significant. New method `ApproxInt_MultiLine::Dump` provides dump of multi-lines. The method `IntPatch_Wline::Dump()` has been modified to show dump of `Wlines` with more precision.
25555	*Summary:* Wrong result of classification of a point relative to solid. It is now checked in `BrepClass3d_Sclassifier` and `BrepClass3d_SolidExplorer` if the point found by Extrema is placed inside the face. This avoids searching for an additional inner point.
25557	*Summary:* Draw command `openoffset` fails on the shape with big values of offset. Trimming of newly created edges has been corrected in some functions of `BrepFill_OffsetWire` class.
25558	*Summary:* It is impossible to restore attached shape. The condition at which the last knot is added has been corrected in method `BsplCLib::PrepareInsertKnots`.
25568	*Summary:* `SIGSEGV` in `thrusections` with edge without 3D curve. `BrepFill_Generator` now sets the algorithm flag `NotDone` if there is a non-degenerated edge with an absent 3D curve.
25578	*Summary:* `SIGSEGV` in `BrepSweep_Rotation` in case of singularities. The check for degenerated edges has been added in method `BrepSweep_Rotation::HasShape`.
25582	*Summary:* `SIGSEGV` in `thrusections` of circle segments. Creating a loft between the two circular edges causes a `SIGSEGV`. Use of trimmed circular curves has been implemented in method `BrepFill_Generator::CreateKPart`.
25584	*Summary:* Wrong result obtained by `PerformInfinitePoint` Test. Periodicity of B-spline surfaces when calculating UV-bounds is now additionally checked in method `BrepTools::AddUVBounds`.
25591	*Summary:* Command mkshell produces wrong shell. New method `BrepLib_MakeFace::IsDegenerated` has been implemented to check whether the edge is degenerated or not. The produced degenerated edges are checked by method `BrepLib_MakeShell::Init`.
25592	*Summary:* Bad result of Fillet operation. The method `Blend_Walking::InternalPerform` has been improved to precise the construction of fillet surface near its boundaries.

25593	*Summary:* Number of intersection points for 2d curves depends on the order of arguments in command `2dintersect`. The command `2dintersect` has been modified to make the results of creation of polygons independent from the order of arguments.
25596	*Summary:* `GCPnts_TangentialDeflection` creates wrong point distribution for visualization. The check of small step after adding a new point has been implemented in method `GCPnts_TangentialDeflection::PerformCurve` to prevent possible jump over the local splash.
25597	*Summary:* Invalid curve on surface in the result of General Fuse operation. The following features have been implemented to improve the results of General Fuse operation ▪ New class `BrepLib_CheckCurveOnSurface` allows calculating the maximum distance between 3D curve and its 2d representation on the face (previously it was done by static methods in `BOPTools_AlgoTools` class). ▪ The method `BOPTools_AlgoTools::ComputeTolerance` and `IntTools_Tools::ComputeTolerance` have been implemented for easy access to this functionality. ▪ The case of Plane/Bspline intersection now can be processed by method `IntTools_FaceFace::ComputeTolReached3d()`. ▪ Method `BOPAlgo_PaveFiller::UpdateFaceInfo` now takes into account new vertices created in PostTreatFF to update Face Information.
25600	*Summary:* Wrong result of Boolean FUSE operation. The algorithm `BOPTools_AlgoTools::FindPointInFace` finding a point in the face now works correctly in the case if the start and projected points are close to each other. The min 3D step has been increased for spherical faces in method `BOPTools_AlgoTools::MinStep3D`.
25614	*Summary:* Provide API access to the new functionalities of Boolean Components. The following methods have been implemented to provide API access to new functionalities of Boolean Components: ▪ `BOPAlgo_Builder::SetArguments` and `BOPAlgo_PaveFiller::SetArguments` set arguments through `TopTools_ListOfShape`. ▪ `BRepAlgoAPI_BuilderAlgo::BRepAlgoAPI_BuilderAlgo` provides object construction using `BOPAlgo_PaveFiller` object. ▪ `BRepAlgoAPI_BuilderAlgo::SetArguments` allows setting arguments. ▪ `BRepAlgoAPI_BuilderAlgo::Arguments()` provides access to the arguments. ▪ `BRepAlgoAPI_BooleanOperation::SetTools` provides access to the tools. ▪ `BRepAlgoAPI_Common::BRepAlgoAPI_Common`, `BRepAlgoAPI_Cut::BRepAlgoAPI_Cut`, `BRepAlgoAPI_Fuse::BRepAlgoAPI_Fuse` and `BRepAlgoAPI_Section::BRepAlgoAPI_Section` provide object construction using `BOPAlgo_PaveFiller` object. Classes `QANewModTopOpe_Glue` and `QANewModTopOpe_Intersection` have become consistent with modifications in `BRepAlgoAPI` package.

25614	The method `BRepAlgoAPI::DumpOper` has been removed. Its functionality is now provided by the file `BRepAlgoAPI_BooleanOperation.cxx` Methods `BOPTest::APICommands` and `BOPTest::OptionCommands` provide the following tcl commands to launch the algorithms implemented in `BRepAlgoAPI` package using `DRAWEXE` application: ▪ `bapibuild r` performs General Fuse algorithm. r is the result of the operation; ▪ `bapibop r type` performs Boolean Operation algorithm: `type` can take the following values: 0 – common, 1 – fuse, 2 – cut, 3 –cut21, 4 – section; ▪ `boptions` dumps the state of current options; ▪ `brunparallel [0/1]` switches between the parallel [1] and sequential [0] mode of computations; ▪ `bfuzzyvalue value` sets the fuzzy value.
25625	*Summary:* Boolean COMMON cannot be built. `BOPTools_AlgoTools2D::Make2D` now performs projection with the tolerance of the edge.
25631	*Summary:* Bounding box is too big for the face. Start and finish indexes of poles have been fixed in case of trimmed Bspline surface in method `BndLib_AddSurface::TreatInfinitePlane`.
25635	*Summary:* Wrong result of 2D-extrema between two ellipses. Lipschitz constant evaluation in case of co-parametrized objects has been corrected in the class `math_GlobOptMin`.
25657	*Summary:* Bad result of Fillet operation. The method `ChFi3d_Builder_C1::Update` has been corrected to obtain correct 2d curves built by fillet algorithm.
25660	*Summary:* `Project` command produces a wrong 2D curve. The starting point of Newton optimization has been changed in method `ProjLib_CompProjectedCurve::Init()`.
25662	*Summary:* `Project` command produces a wrong 2D curve. The concatenation algorithm in class `ProjLib_ComputeApproxOnPolarSurface` has been fixed to work with periodic Bspline surfaces.
25663	*Summary:* Expand `math_PSO` class description. The description of class `math_PSO` has been extended with information about “step by step” algorithm, which helps to determine its domain of applicability.
25692	*Summary:* Improvement of Boolean operations. The method `IntPolyh_MaillageAffinage::TriContact()` now avoids creating local objects for detailed analysis if the algorithm exits by simple conditions checked at the beginning.

25701 25677	*Summary:* Problem with fillet symmetry on two perpendicular cylinders. Classes `ChFi3d_Builder` and `Blend_Walking` now bind the boundaries of the fillet surface with boundaries and geometric form of the initial shape.
25704 25705	*Summary:* `BRepOffsetAPI_MakeOffset`: incorrect processing. `MakeOffset` function from `BrepFill_OffsetWire` now enlarges offset edges as much as possible in `GeomAbs_Intersection` mode to provide their intersection.
25708	*Summary:* `GeomAPI_ExtremaCurveCurve` does not return all intersection points. The class `Math_GlobOptMin` has been improved to expand coefficients between neighboring indexes and to change the starting condition of local optimization.
25711	*Summary:* Get rid of static variable `islambdadefined` in `AppParCurves_BspGradient`. The static variable `islambdadefined` has become a non-static field member in class `AppParCurves_BspGradient`.
25715	*Summary:* Intersection between cylinders produces excess vertices. New function `IsSeamOrBound` from class `IntPatch_Intersection` allows checking if a point lies on a seam-edge (if exists) or surface boundaries.
25718	*Summary:* Unstable work of tests. Stable work of the algorithm `BrepAlgo_Loop` has been provided by replacing `DataMap` collections with `List` and/or `IndexedDataMap`, which makes iterations independent on the memory addresses. All fields have been initialized in the default constructor of `Extrema_PonSurf`.
25719	*Summary:* Boolean operations crash. Class `math_BFGS` has been modified to avoid calling `Perform()` in its constructor. It must be called explicitly.
25720	*Summary:* Incorrect code of math classes can lead to unpredicted behavior of algorithms. The constructors & destructors of classes `math_BissecNewton`, `math_BrentMinimum`, `math_FRPR`, `math_FunctionSetRoot`, `math_NewtonFunctionSetRoot`, `math_NewtonMinimum` and `math_Powell` have been modified to avoid calling virtual methods.
25721	*Summary:* Wrong result obtained by Common operator. Processing of internal edges has been added in method `BOPTools_Set::Add`.
25729	*Summary:* `BrepOffset_MakeOffset()` produces wrong result for join type Intersection. Functions `Correct2dPoint` and `DefineConnectType` have been added in class `BrepOffset_Analyse` to improve processing of intersection cases by the offset algorithm.

25735	*Summary:* Wrong solid is considered as valid by `checkshape`. Processing of shells with `INTERNAL` faces has been added in method `BrepCheck_Solid::Minimum()`.
25742 26009	*Summary:* Partition of 2 shapes stresses a performance issue. The following changes have been implemented to improve the performance of partition algorithm in class `IntPatch_ImpImpIntersection`: ▪ The algorithms computing `aStepU1` and a point in V-boundaries computing have been improved. ▪ The situation when the intersection line walks along V-boundary of a cylinder is processed better. ▪ The intersection lines are created with their individual step along U1 parameter. ▪ Processing of points has been moved to the assembly level. The interface to convert `gp_XY(Z)` has been added in class `math_Vector`. New option `[-d]`, which allows storing the intermediate result of the operation, has been added for commands `bfuseblend` and `bcutblend`.
25746 25794	*Summary:* Excessive memory use in `math_Matrix`. The class `math_DoubleTab` has been modified to statically allocate only 16 items for local buffer. Indirection table has been removed from `math_DoubleTab`.
25757	*Summary:* `Distmini` returns wrong solution for ellipse/vertex. Analytical handling of degenerated cases has been added in method `math_TrigonometricFunctionRoots::Perform`.
25766	*Summary:* Exception in `BrepAlgo_Section`. Protection from null `wlines` has been added in method `BrepAlgo_Section::MergeWLinesIfAllSegmentsAlongRestriction`.
25769	*Summary:* Replace `BOPCol_Array1` with `NCollection_Vector`. New method `NCollection_BaseVector::SetIncrement` allows setting the size of increment dynamically (not in the constructor). Class `BOPCol_Array1` has been removed. Classes from `BOPDS` package have been modified accordingly.
25780	*Summary:* `Checkshape` raises an exception `Standard_OutOfMemory`. Memory leak has been fixed in method `BrepCheck_Wire::Propagate`.
25782	*Summary:* The result of intersection between two cylinders is incorrect. The algorithm computing the intersection line (in case of cylinders with two parallel axes) has been changed in method `IntAna_QuadQuadGeo::Perform`.
25788	*Summary:* Parallelization of BOP Builder algorithm on the second level. Method `BOPAlgo_Builder::FillIn3Dparts` has been modified to provide parallel processing.

25810	*Summary:* Exception in extrema operation. The algorithm computing the number of analytic intersection points has been corrected in method `Extrema_ExtElCS::Perform`.
25818	*Summary:* `IntTools_FaceFace` throws `Standard_ConstructionError` with two unbounded planes. Processing of infinite planes has been corrected in method `IntTools_FaceFace::Perform`.
25819	*Summary:* Bad result of BOP cut on valid shapes. The algorithm adjusting parameter to the surface boundaries has been changed in class `IntPatch_ImpImpIntersection`. `First` and `Last` parameters are controlled in function `BrepLib_CheckCurveOnSurface::Compute()`.
25828	*Summary:* `BRepAlgoAPI_Section` fails for a solid and a face depending on order of arguments. Additional analysis is performed in the class `IntPatch_PrmPrmIntersection` to decide if it is necessary to reject the current line. If Walking-lines coincide, the longer line or the one containing more points is kept (because it is most likely to be more precise).
25838	*Summary:* Wrong result obtained by General Fuse operator. Method `BOPTools_AlgoTools::GetFaceOff` has been modified to compare angles using the angular tolerance value.
25841	*Summary:* Incorrect edge displaying. Handling of cases when the chordal deviation of the displayed edge is much greater than the deflection has been improved in methods `GCPnts_TangentialDeflection::PerformCurve` and `BrepMesh_EdgeTessellator::BrepMesh_EdgeTessellator`.
25842	*Summary:* Wrong intersection 2D-curves obtained for pair of faces. Handling of degenerated parametrization (such as sphere) has been added in method `IntWalk_Iwalking::ComputeCloseLine`.
25847	*Summary:* Wrong result obtained by General Fuse operator. The following improvements have been implemented in General Fuse operator: ▪ Intended set of VE interferences has been refined in method `BOPAlgo_PaveFiller::PerformEE()` to avoid unwanted overlaps of pave blocks. ▪ Increment of VE and VF interferences can be initialized in methods `BOPAlgo_PaveFiller::ForceInterfVE` and `BOPAlgo_PaveFiller::ForceInterfVF` correspondingly.

25858	*Summary:* Incorrect result of open offset on single edge based on Bspline curve. A particular case of single Bspline curve is now properly processed in function `KpartCircle` from `BrepFill_OffsetWire`.
25861	*Summary:* Wrong result obtained by projection algorithm. Handling of trimmed analytical surfaces has been added in method `Extrema_ExtPS::TreatSolution`.
25876	*Summary:* `Geom2dAPI_InterCurveCurve` returns only one intersection point instead of two intersection points. The domain of circle has been extended in method `IntCurve_IntConicConic::Perform` to include all possible solutions.
25880	*Summary:* Fuzzy Boolean operations fail with multiple tools. New method `BOPTools_AlgoTools2D::AttachExistingPCurve` has been added to attach P-Curve on face from one edge to another. This mechanism is now used for processing of E/E common bocks in method `BOPAlgo_PaveFiller::MakePCurves()`.
25883	*Summary:* `BRepOffsetAPI_MakePipeShell` produces invalid result. The field `myMaxSegments` is now initialized in the class `BrepFill_PipeShell` in the same way as in constructor of `BrepFill_Pipe`.
25886	*Summary:* Wrong result obtained by projection algorithm. The current iteration approximation is now used in method `Approx_ComputeCLine::Perform` if necessary.
25887	*Summary:* Invalid pipe construction. Recognition of a particular cylindrical has been added in local function `IsSweepParallelSpine` from `GeomFill_Sweep`.
25890	*Summary:* Intersection algorithm produces overlapped curves. New function `IntImp_Int2S::ChangePoint()` has been implemented to return the intersection point enabled for changing. The walking algorithm tries to forbid breaking `WLine` if it goes along surface boundary.
25892	*Summary:* Wrong result obtained by projection algorithm. The following improvements have been implemented in the projection algorithm: ▪ Method `ProjLib_CompProjectedCurve::Init()` now can process the case when the algorithm of projection to Bspline surface returns a null 2d curve; ▪ Approximation of the next step has been changed in the default projection algorithm; ▪ Special handling of a surface of revolution has been added in method `ProjLib_ProjectedCurve::Load`.

25894	*Summary:* `BRepOffsetAPI_NormalProjection` fails to projection an edge on a face. Method `ProjLib_CompProjectedCurve::Init()` has been fixed to improve processing in case of trimmed parameters.
25898	*Summary:* Wrong result obtained by intersection algorithm. The walking algorithm has been improved in method `IntWalk_Pwalking::Perform`.
25908	*Summary:* `BSpline` (U2 < U1) in `HLRBRep_HLRToShape.` The hidden line removal algorithm has been improved to provide correct projection.
25937 25941	*Summary:* Failed Cut Edge or Face by HalfSpace. The following modifications have been introduced to improve cutting of edges or faces by HalfSpace: ▪ New method `IntTools_Context::BndBox` returns a reference to the bounding box of the shape. ▪ New method `IntTools_Context::IsInfiniteFace` returns true if the face has infinite bounds. ▪ Methods `BOPAlgo_BuilderFace::PerformAreas()` and `BOPAlgo_BuilderSolid::PerformLoops()` now include unrestricted faces in the list of areas and loops correspondingly.
25938	*Summary:* `BRepBuilderAPI_Transform` is not thread safe. The method `BRepBuilderAPI::ModifiedShape()` now returns a shape by value, not by reference.
25951	*Summary:* Exception in intersection operation. Processing of overlapping curves has been improved in method `IntPatch_PrmPrmIntersection::Perform`.
25953	*Summary:* Variable `BRepMesh_PairOfIndex::Prepend` assigned twice. Index assignment has been corrected in method `BrepMesh_PairOfIndex::Prepend`.
25954	*Summary:* Formatting/logic mismatch in `GeomPlate_Surface::SetBounds`. A sequence of condition statements has been expressed more logically in method `GeomPlate_Surface::CallSurfinit().`
25969	*Summary:* Wrong result obtained by 2D classifier algorithm. The method `IntTools_Fclass2d::Init` now checks if derivative angles are near to PI and avoids express treatment of such wires.
25976	*Summary:* Conversion of an offset face from a spline into a spline face hangs-up the application. The algorithm of offset surface approximation has been improved in method `GeomConvert::SplitBSplineSurface`.

25980	*Summary:* Wrong result obtained by projection algorithm. The algorithm `ProjLib_CompProjectedCurve::Init()` now takes into account the cases when projection cannot be performed for non-analytical and non-spline surfaces.
25982	*Summary:* Wrong result obtained by General Fuse operator. New method `BOPTools_AlgoTools2D::AdjustPCurveOnFace` uses the reference to `BrepAdaptor_Surface` object as parameter to avoid redundant computations such as `BrepTools::UVBounds()`.
25992	*Summary:* `Method Geom2dAPI_InterCurveCurve::Segment()` requires a revision. The method `Geom2dAPI_InterCurveCurve::Segment()` has been has been fixed to return correct 2D curves in many cases.
26008	*Summary:* Wrong intersection 2D-curves obtained for pair of faces. The method `ProjLib_ComputeApprox::Function_SetUVBounds` has ceased to adjust to the interval boundaries the points of 2D curve found by the intersection algorithm.
26016	*Summary:* `BRepPrimAPI_MakeRevol` crash. Processing of the case when the line of a body of rotation is a trimmed curve has been corrected in method `BrepSweep_Rotation::SetGeneratingPCurve`.
26038	*Summary:* Wrong result done by extrema for the circle and plane. The Extrema algorithm has been corrected in class `Extrema_ExtElCS` for the case of intersection between a circle and a plane.
26041	*Summary:* `GeomLib_Tool::Parameter` method fails. The formulas for calculation of hyperbola have been improved in class `GeomLib_Tool`.
26064	*Summary:* `Distmini` of two edges locks up. The method `Extrema_GenExtCC::Perform()` has been refactored to avoid an inefficient algorithm of duplicate points removal. Instead, duplications are checked when new points are added. Fields are now initialized in constructors of the class `Extrema_GenExtCC`; unused classes `Extrema_ELCC` and `Extrema_ELCC2d` have been removed.
26073	*Summary:* Correction for Canonical Recognition product. A case of empty handle usage has been fixed in class `BrepTopAdaptor_TopolTool`.
26080	*Summary:* Wrong result obtained by General Fuse operator. The minimum of two available values is now used in class `IntTools_Fclass2d::Perform` for classification.

26098	*Summary:* The result of General Fuse operation is a self-interfered shape. The method `BOPTools_AlgoTools::IntersectCurves2d` now checks the validity of 2D intersection before applying the result.
26112	*Summary:* Exception is raised during perform of General Fuse operation. The method `BOPAlgo_WireSplitter::RefineAngle2D` has been protected against null vector.
26118	*Summary:* Implement fast sewing algorithm. A special “fast sewing” algorithm, which is much faster than the general-purpose sewing, has been implemented in class `BRepBuilderAPI_FastSewing` and Draw command `fastsewing`. The fast sewing algorithm works as follows: ▪ Since all sewn faces use natural restriction, only a list of surfaces `Geom_Surface` is required at input. ▪ If the information about neighbors for each patch is available for your application, then it can be used to join faces. If not the algorithm restores this information from a flat list of surfaces by geometrical comparison of some points on patch borders. ▪ For each surface in the list, the algorithm directly constructs 4 valid edges that join it with the neighbor faces. New method `BrepLib::EnsureNormalConsistency()` provides an algorithm allowing to ensure that shape triangulation has consistent normals at the connections between faces. The algorithm computes all normals, stores them in `Poly_Triangulation` objects in the faces, and checks the edges between each two faces. If the angle between normals at the same point is less than the threshold (0.1° by default) the normal is replaced by average. This functionality is accessible in Draw via command `correctnormals`.

Visualization

23200 26014	*Summary:* Prevent multiple triangulating of a shape that already has been triangulated. New flag `IsAutoTriangulated` has been added to `Prs3d_Drawer`. If this flag is `True` (by default) automatic re-triangulation will be applied with deflection-check logic. If it is disabled, the application can handle triangulation by itself avoiding possible side effects of automatic procedure. New parameter `-autoTriang` has been added in `vdefaults` command to control the new functionality within Draw.
23484 25475 25804	*Summary:* `TKOpenGl` – primitive arrays to become the only way to render geometry. The packages `TKOpenGl` and `TKV3d` have been revised to make primitive arrays (hardware-accelerated when VBO support is available) the only way to render geometry The old code drawing some objects (e.g. a capping plane or a trihedron in wireframe mode) without using primitive arrays has been removed.
24623 25933 26031 26115 26121 26139 26146 26147	*Summary:* Redesign of selection mechanism. New efficient selection algorithms including a 3-level BVH tree and calculation of selection in 3D space have been implemented for 3D selection instead of the old resource-consuming mechanism based on software projection of sensitive entities onto the current screen plane (updated on each rotation). The following modifications have been introduced: ▪ Intersection checks have been moved to `SelectMgr_BaseFrustum` descendants; ▪ `SelectMgr_ViewerSelectors` are now shared between local and global contexts; ▪ Transformations of sensitive entities are now stored in `SelectMgr_SelectableObject` only. Sensitive entities are independent from transformations, which are applicable to `SelectMgr_SelectingVolumeManager` instance only; ▪ Connected and multi-connected interactive objects are now represented by their child objects only for `SelectMgr_SelectionManager`; ▪ If an interactive object has child objects, they are now stored as separate objects in `SelectMgr_SelectionManager`. Two modes of rectangular selection are now available: selection of only the objects that are fully included in the rectangle and selection of all objects overlapped by the triangle. The method `StdSelect_ViewerSelector3d::AllowOverlapDetection` allows switching between these modes. The corresponding option `-allowoverlap` has been added to command `vselect`. In Draw, the interactive rectangular selection is available in 2 modes: ▪ If the user starts selection from upper corners, only fully included objects are selected; ▪ If the user starts selection from lower corners, both partially and fully overlapped objects are selected. See also Porting to version 6.9.0 section for some useful recommendations.

24934	*Summary:* Implement a more general way for rendering of immediate objects. The application is now able to define its own immediate objects in neutral context by assigning them to a predefined Z-layer.
25091 25897 25851 25984 26165	*Summary:* Use FBO for layer with immediate objects. The new implementation of immediate mode (for temporary and highly-interactive objects) based on FBO usage eliminates possible artifacts caused by the previous implementation and allows using this functionality on a system without the possibility to draw directly into front buffer (e.g. Android).
25136	*Summary:* Fix multiple `Aspect_ColorScale` usage issues. The following changes have been introduced in class `Aspect_ColorScale`: ▪ The methods `SetColor`() and `SetLabel`() now work with 0-based index. ▪ Method `SetColor`() now checks the length of `myColors`. ▪ Methods `GetCurrentColor`() and `GetCurrentLabel`() now can be used to get user-specified and default colors / labels. ▪ `DrawScale`() – shows labels even for one interval
25304 25328 25474 25500 25539 25580 25710 25854 26002 26004 26012	*Summary:* OpenGL ES 2.0 and OpenGL 3.2+ core profile. Text rendering, object texturing, environment map texturing within built-in GLSL programs have been implemented. Global trihedron presentation has been fixed. OpenGL parameters stack usage has been eliminated. Misprints in detection of high precision floats within OpenGL ES 2.0 have been fixed. The new option `OpenGl_Caps::contextCompatible` allows requesting either compatibility or core OpenGL profile (compatibility profile is requested by default).
25372	*Summary:* `TKOpenGl` – suppress annoying verbose messages from NVIDIA OpenGL driver OpenGL package has been revised drop functions unrelated to OpenGL 4.2 core functionality.
25436	*Summary:* `AIS_InteractiveContext::HilightPreviousDetected()` should switch from first value in the list to the last. Method `AIS_InteractiveContext::HilightPreviousDetected()` now properly returns to the last detected entity after reaching the first one.
25459	*Summary:* `AIS_ColoredShape::SetMaterial()` should not reset custom colors. The custom color assigned to sub-shapes within `AIS_ColoredShape::SetMaterial()` method is now preserved.
25466	*Summary:* Impossible to change the display mode when a local context is opened. Irrelevant check has been removed from method `AIS_InteractiveContext::SetDisplayMode()`.

25483	*Summary:* `TKOpenGl` – fix memory leak due to unused stack in `OpenGl_StateInterface`. Memory leak caused by unused stack in `OpenGl_StateInterface` has been fixed throughout the `OpenGl` package.
25484	*Summary:* Group sub-shapes with the same style in `XCAFPrs_AISObject::Compute()`. Performance regression arising after migration of `XCAFPrs_AISObject` class to `AIS_ColoredShape` has been fixed.
25492	*Summary:* The selected subshape does not have topological relationship with the original shape. The method `SelectMgr_EntityOwner::SetLocation()` has been fixed to return identity owner location if the selectable object has identity transformation.
25507	*Summary:* The method `V3d_View::Place()` is incorrect. The incorrect result of method `V3d_View::Place()` has been fixed. The y coordinate passed to `Pan()` method has been replaced by `(height - y)`.
25511	*Summary:* Drop redundant viewer option `V3d_View::Transparency()`. Confusing viewer option `V3d_View::Transparency()` has been removed. `Visual3d_ViewManager` now activates texturing by default.
25528	*Summary:* Exception on removing an interactive object from a local context. The method `AIS_LocalContext::Remove()` has been corrected to unbind the argument from `myActiveObjects`.
25532	*Summary:* Fix cross-references between `AIS_ConnectedInteractive` and connected presentation. The method `AIS_ConnectedInteractive::Connect()` has been fixed to properly release memory after AIS_Interactive_Context::RemoveAll().
25540 25651	*Summary:* Discretization of the circle differs in shaded and wireframe modes. The following modifications have been introduced to improve the visualization of circle discretization: ▪ `HilightDrawer()` method has been added to `SelectMgr_SelectableObject`. Its value replaces static drawer in `StdSelect_Shape`. ▪ Computed relative deflection is stored as absolute deflection and can be used for sub-shapes. ▪ `AIS_ColoredShape` object now uses for sub-shapes the relative deflection computed for the main shape.

25544	*Summary:* TKOpenGl – support grayscale textures. The following improvements have been implemented to support grayscale textures: ▪ Enumeration Image_PixMap::ImgFormat has been extended by ImgAlpha and ImgAlphaF. ▪ GL_REPLACE is specified in OpenGl_Workspace::setTextureParams() for one-component textures with disabled modulation. ▪ OpenGl_Texture::GetDataFormat() returns GL_LUMINANCE format for ImgGray format and GL_ALPHA for ImgAlpha. ▪ Command vmarkerstest now uses ImgAlpha instead of pixel format for grayscale images.
25552	*Summary:* Provide the way to hide the object in specified view of the viewer. It has become possible to hide an object in the specified V3d_View without creation of a dedicated V3d_Viewer. For this purpose: ▪ The number of Views within a single Viewer instance has been limited to 32 in Visual3d_ViewManager. ▪ New method AIS_InteractiveContext::SetViewAffinity() has been added to define object affinity and to hide object within specific views.
25611 25974	*Summary:* Improve design and implementation of graduated trihedron. The interference caused by simultaneous display of Z-buffer and graduated trihedrons has been fixed. The following modifications related to visualization of graduated trihedron have been introduced: ▪ Method V3d_View::GetGraduatedTrihedron() now returns configuration structure directly instead of a long arguments list. ▪ Global variables have been eliminated in method OpenGl_Trihedron::Setup().
25624	*Summary:* Selection is incorrect in perspective mode in a specific case. The scaling for perspective projection matrix is now done by class Select3D_Projector. The computation of picking line for perspective camera has been corrected.
25627	*Summary:* SelectedShape() and HasSelectedShape() of AIS_InteractiveContext class do not work as expected. The algorithm calculating location in method AIS_InteractiveContext::SelectedShape() has been corrected. New method AIS_LocalContext::HasSelectedShape() does not take shape decomposition into account.
25650	*Summary:* AIS_Line::ComputeSelection() should not ignore the selection mode argument. From now on, the method AIS_Line::Compute() supports only selection mode 0.

25652	*Summary:* TKOpenGl – Ray Tracing initialization failures are not properly reported. Logging of GLSL warnings have been added in ray-tracing core to allow providing sufficient information about Ray-Tracing initialization failures.
25661	*Summary:* AIS_InteractiveContext::Load() is not symmetric to the local context. The method AIS_InteractiveContext::Load() now registers theIObj in the selection manager to prepare further activation of selection.
25664	*Summary:* Dynamic highlighting should not be discarded on re-displaying independent object. The list of immediate presentations is now properly cleared in AIS_LocalContext::manageDetected() instead of AIS_LocalContext::Unhilight().
25671 25672	V3d_View::Convert doesn't work as expected in GRID active mode. The conversion of coordinates to grid has been removed from methods V3d_View::Convert and ::ConvertWithProj. The following draw commands have been added: ▪ vconvert for testing the conversion methods; ▪ vprivilegedplane for setting/printing the coordinate system of the grid plane.
25675	*Summary:* Fix problems and inefficiencies with frustum culling. Unnecessary overlap check of layer items has been removed in method OpenGl_Layer::traverse(). The calculations in overlap detection methods in class OpenGl_BVHTreeSelector have been optimized.
25679	*Summary:* TKOpenGl – View frustum culling clips wrong objects. The algorithm calculating area for degenerated bounding boxes has been corrected in class BVH_Box.
25687	*Summary:* XCAF – eliminate visual artifacts at the edges of faces. The following improvements have been introduced to eliminate visual artifacts: ▪ The algorithm splitting input shapes into Closed and Open volumes has been improved in method AIS_ColoredShape::Compute() enabling back-face culling whenever possible. ▪ The last argument of method StdPrs_ShadedShape::Add() has been changed from Boolean to enumeration StdPrs_Volume, which controls parsing of Closed/Open volumes within the input shape. ▪ The method StdPrs_ShadedShape::ExploreSolids() has become public. ▪ Draw command vaspects has been provided with option –setvisibility to hide subshape (using AIS_ColoredShape).
25691	*Summary:* TKService – fix font corruption in FreeType 2.5.4. The method Font_FTFont::RenderGlyph() has been fixed to avoid artifacts caused by FreeType version 2.5.4.

25703 25762	*Summary:* TKOpenGl – Decrease number of samplers used in ray-tracing mode. The samplers myObjectNodeInfoTexture, myObjectMinPointTexture and myObjectMaxPointTexture have been removed. Serialized data contained in texture buffers has been added to global scene buffers: mySceneNodeInfoTexture, mySceneMinPointTexture and mySceneMaxPointTexture correspondingly.
25723	*Summary:* TKV3d – the center of rotation should be calculated taking into account structure visibility and selection flags. V3d_View::Gravity() computes the bounding box of a scene only including highlighted objects, if there are any, to provide a context-oriented center of rotation. Draw command vrotate has been extended with new flags –mouseStart and –mouseMove, which emulate rotation by mouse.
25732	*Summary:* TKOpenGl – back face culling should not affect textured font rendering. The method OpenGl_TextFormatter::Result has been corrected to orient triangles in normal counter-clockwise order (GL_CCW).
25758	*Summary:* TKOpenGl – texture initialization fails on Intel HD 4600 in ray tracing test. The method OpenGl_Workspace::DisableTexture() has been modified to unbind global texture sampler right after its usage.
25760 25768	*Summary:* Graphic3d_Structure – Stop using invalid bounding boxes of empty groups. The following improvements have been introduced to avoid use of invalid bounding boxes of empty groups: ▪ Creation of an empty group is avoided in Prs3d_WFShape::Add(); ▪ Uninitialized bounding box is not used in Graphic3d_Structure::minMaxCoord(). ▪ The precision factor in method Graphic3d_Camera::ZFitAll has been tweaked to avoid hiding objects on a near or far plane in some cases.
25773 25955	*Summary:* Remove AIS_Drawer class and transfer its Link() logic to Prs3d_Drawer. AIS_Drawer class has been removed and all its logic was transferred to Prs3d_Drawer. Drawer object and methods SetAttributes(), Attributes() and UnsetAttributes() have been transferred from AIS_InteractiveObject to SelectMgr_SelectableObject. Link() is not auto-created, so its attributes should not be used before setting of Link to Drawer (before Display call by default). It is necessary to set custom aspects explicitly with SetApsectX methods. If attributes are not set, the default value or value from Link() will be used. The method ClearLocalAttributes() removes all own attributes. Draw Harness command vaspects has been extended by the parameter –defaults. If this flag is specified, the presentation properties will be assigned to all objects that have not their own specified properties and to all objects to be displayed in the future.

25783	*Summary:* Allow Z-layer to draw 2D objects and to make it alternative to Overlay and Underlay Displaying objects in 2D now works using Z-layers and transform persistence. The following corresponding features have been implemented: ▪ Pre-defined Z-layer `Graphic3d_ZlayerId_BotOSD` has been added to draw the underlay. ▪ Transformation persistence flags `Graphic3d_TMF_2d_IsTopDown` and `Graphic3d_TMF_2d` have been defined for displaying objects in screen coordinates. ▪ Method `BeginTransformPersistence()` has been updated to transform matrices for drawing 2d objects. ▪ Anchor point defines angle of the screen to display the object in similar way as for `Graphic3d_TMF_TriedronPers`. Draw command `vdisplay` has been extended with new options: ▪ `-overlay` and `–underlay` to display objects in overlay and underlay; ▪ `-select` and `–noselect` were added to control selection; selection for 2d objects is turned off by default; ▪ `-tpposition` and `-tppos` was added to set a translation point for transform persistence. ▪ `-dispMode` and `–highMode` to define displaying and highlighting modes; ▪ `-2d` and `3d` to display object in 2d or 3d; ▪ `-2dtopdown` to make the Y axis of the view point down (for 2d objects); ▪ `-transpers/-tps` to set a transform persistence mode for the object.
25788	*Summary:* Gradient background will cut view if there is `ClipPlane` defined. The method `OpenGl_View::DrawBackground` now renders the capping plane with a primitive array.
25800	*Summary:* `TKOpenGl` – disable `GL_DITHER` explicitly. The obsolete feature `GL_DITHER` has been disabled by default.
25809	*Summary:* `TKOpenGl` – fix texture mapping in capping. The capping plane coordinates have been reversed to preserve a natural texture orientation (e.g. to make text readable).
25814	*Summary:* `Prs3d_WFShape::AddPolygon()` – always use polygonal representation from edge regardless from requested deflection. The method `Prs3d_WFShape::AddPolygon()` has been modified to always use polygonal representation from edge regardless of the requested deflection.
25815	*Summary:* Error message if texture loading fails. The error message is now shown if the texture cannot be loaded from file. The corresponding messages have been provided in method `Image_AlienPixMap::Load`.

25822	*Summary:* TKOpenGl – front material should be used instead of back material in GLSL when distinguish mode is turned off. The method OpenGl_ShaderManager::PushAspectFace() now uses front material instead of back material in GLSL when distinguishing mode is turned off.
25824	*Summary:* TKV3d – skip infinite presentations when computing gravity center of the view scene. The method V3d_View::Gravity() has been modified to skip infinite presentations when computing center of the view scene for interactive rotation.
25833	*Summary:* Ray Tracing – fix problems with the backside of triangles. The calculation of reflections on the backside of triangles has been corrected by implementing a two-sided lighting model. Ray-tracing shader has been optimized (up to 25% performance increase).
25867	*Summary:* TKOpenGl – fix misprint in transformation persistence math. A misprint has been fixed in method OpenGl_Utils::Project().
25885	*Summary:* Ray tracing – Improve layer processing. The ray-tracing core has migrated from OpenGl_Workspace to OpenGl_View.
25893	*Summary:* TKOpenGl – do not use uninitialized memory to track Clipping Planes state. The method OpenGl_View:Render has been modified to avoid using uninitialized memory to track the state of Clipping Planes.
25906	*Summary:* V3d_View::Rotation() eliminate erroneous viewer redraw within disabled immediate update. The redundant view update of V3d_View::Rotation method has been removed.
25931	*Summary:* Possibility to initialize an environment texture by Image_PixMap instance. It has become possible to initialize Graphic3d_TextureEnv object with an instance of Image_PixMap class.
25935	*Summary:* TKV3d, Exception when displaying shell in the viewer. The processing of double precision numbers has been fixed in method Visual3d_View::MinMaxValues.
25964	*Summary:* TKOpenGl – compile RayTracing shader without texturing when no textures in use. USE_TEXTURES code paths are now disabled even when bindless textures are supported by driver.
25973	*Summary:* TKOpenGl – support EAGLContext as alternative to NSOpenGLContext. Compatibility with iOS has been provided.

25978	*Summary:* Setup font aliases for Android. System fonts `Droid Sans Mono`, `Droid Serif` and `Roboto` have been implemented as Android aliases to `Courier`, `Times new Roman` and `Arial` in class `Font_FontMgr`. These Android system fonts have been added in directory `/system/fonts`. The method `OpenGl_Text::FindFont()` now prints an error message if some fonts are missing. The method `OpenGl_Text::render()` allows straightforward font rendering on OpenGL ES.
26025	*Summary:* `TKOpenGl` – stereoscopic output does not work. The following modifications have been implemented to improve the stereoscopic output: ▪ The target FBO is passed as parameter in method `OpenGl_View::Render()`. ▪ Read/Write buffers management logic has been revised in class `OpenGl_Context` taking into account FBOs. ▪ `Lprojection` and `Rprojection` are now set up in `Graphic3d_Camera::UpdateProjection()` class in the same way as `Mprojection` in case of `Projection_MonoLeftEye` / `Projection_MonoRightEye` to provide API consistency.
26029	*Summary:* Poor performance of connected objects The problem with performance of `SelectMgr_SelectableObjectSet` has been fixed.
26070	*Summary:* Ray tracing with reflections is poor on rotated presentation. The problem with incorrect generation of secondary rays in case of non-identity viewer transformation applied to `AIS` shape has been fixed by handling such transformations in ray-tracing shaders.
26076	*Summary:* Empty bounding box of a shape after closing local context. Handling of invalid bounding boxes has been fixed in method `AIS_InteractiveObject::BoundingBox.`
26081	*Summary:* `TKOpenGl` - rebuild vertex attributes in order to not render large index arrays in OpenGL ES. To provide compatibility with OpenGL ES 2.0 devices, which do not support index arrays larger than 64K elements, class `OpenGl_PrimitiveArray` has been modified to drop the index buffers above the supported threshold and to duplicate the vertex attributes instead.
26109	*Summary:* Add method `ChangeAxisAspect(int)` to `Graphic3d_GraduatedTrihedron.` New method `Graphic3d_GraduatedTrihedron::ChangeAxisAspect` allows setting axis parameters.

26120	*Summary:* Segmentation fault in `AIS_Selection`. NULL-pointer checks have been added in methods `AIS_Selection::ClearAndSelect` and `AIS_Selection::Single()`.
26128	*Summary:* `TKOpenGl` - fix misprint in external GLX context initialization. The warning about missing caps in window `Visual` has been added in class `OpenGl_Window`. Initialization of an alien GLX context is now allowed.
26159	*Summary:* Revise tolerance implementation for selection. Max tolerance is now applied to all objects and entities by default. If there is an entity with a lower tolerance, selecting frustum will be recalculated for it. New function `AIS_InteractiveContext::SetPixelTolerance` disables the mechanism of adaptive tolerance calculation implemented in `SelectMgr_ViewerSelector` and sets the given tolerance for ALL activated sensitive entities. New method `SelectBasics_SensitiveEntity::SetSensitivityFactor` allows managing the sensitivity of each entity individually.
26172	*Summary:* `AIS_LocalContext` - locally selected object should not stay in the viewer after deactivation in the local context The method `AIS_LocalContext::ClearOutdatedSelection()` has been fixed to deselect the entities which belong to a deactivated mode.

Application Framework

25394	*Summary:* Store/retrieve the list-based attributes containing no items. It has become possible to store/retrieve list-based attributes containing no items.
25501	Summary: `Tnaming::Displace` calls itself recursively with default parameter. A misprint has been fixed in method `Tnaming::Displace()`.
25524	*Summary:* `XmlTObjDrivers_ModelDriver::Paste` causes crash when saving a model. A misprint has been fixed in method `XmlTObjDrivers_ModelDriver::Paste`.
26006	*Summary:* `Backup()` is not efficient in `TDataStd` array attributes The implementation of `Backup()` methods of the standard array attributes has been changed to decrease memory consumption.
26061	*Summary:* `Tnaming_Selector` crash in select method. `Tnaming_Localizer` has been protected against unexpected input arguments.
26155 26157	*Summary:* `Tnaming, CurrentShape:` order of shapes in Modification compound is unpredictable `TopTools_MapOfShapes` has been replaced with `TopTools_IndexedMapOfShape` in `Tnaming` package to get a fixed order of shapes in resulting compounds.

Data Exchange

23328	*Summary:* Importing VRML files with `scaleOrientation` not possible. Recognition and interpretation of key-words `scale` and `scaleOrientation` have been corrected in method `VrmlData_Group::Read`.
23800 25632	*Summary:* IGES writer loses face orientation. The method `BRepToIGES_BRShell::TransferFace` now reverses surfaces for writing faces with a reverse orientation.
24601	*Summary:* Unwanted spheres shown after Step-Import. The method `StepToTopoDS_TranslateFace::Init` has been modified to check for the outer boundary before creating a wire from Vertex Loop on spheres.
25176	*Summary:* STEP Reader – no error report if the referenced entity has a wrong type. The method `Interface_CheckTool::CompleteCheckList()` is corrected to not reset the Check added to `CheckList`.
25275	*Summary:* Different result of reading operation from `*.igs` and `*.stp` file for WINDOWS and LINUX platforms. Symbol `SUB` (ASCII-code `0x1A`) is now considered end-of-file on both WINDOWS and LINUX systems.
25279	*Summary:* OCCT fails to read VRML file created by OCCT. The following improvements have been introduced in processing of VRML files: ▪ The parameter corresponding to VRML format version has been added to `VrmlAPI_Writer::Write()` and `VrmlAPI::Write()` to allow using both versions of VRML by the same writer. ▪ The command `writevrml` now can write VRML files of both versions v1.0 and v2.0, in wireframe or shaded mode, or both. Useless parameter `Deflection` has been removed. ▪ Meshing has been removed from writers of both versions. Shaded representation is skipped if a mesh does not exist. Wireframe representation checks the existence of a mesh before. If the mesh exists, deflected edges are taken from the mesh; otherwise, they are generated with default deflection. ▪ Drawing of redundant edges has been removed in wireframe representation of VRML version 1.0 (mesh on non-planar surfaces does not match real edges of `TopoDS_Shape` or the representation in version 2.0).
25357	*Summary:* STL writer does not check the given shape for existing triangulation and meshes it again using `BrepMesh` in force mode. Meshing functions have been completely removed from `StlTransfer`. Now `StlWriter` can return error status, for example, if a mesh of the passed shape is empty.

25518	*Summary:* Colors are not written to IGES 5.3. Writing of colors to IGES Face (510) and Solid (186) entities has been enabled in method `IGESCAFControl_Writer::MakeColors` since this feature is used by CAD systems (even if not allowed by IGES standard).
25523	*Summary:* Fail to read back solid written to STEP. A step file reading exception has been fixed in method `StepToTopoDS_TranslateEdgeLoop::RemoveSinglePCurve.`
25634	*Summary:* Checking of compliance of vertices and pcurve fails. The method `ShapeAnalysis_Edge::CheckVerticesWithPCurve()` has been modified to properly take the location of a face into account.
25689	*Summary:* `IGESCAFControl_Writer` crash in constructor. `IGESData_BasicEditor` is now initialized after the initialization of `IGESControl_Controller`, so that `BasicEditor` could always find template "`iges`". Additionally, `IGESData_BasicEditor` now can be initialized via `Init()` method.
25694	*Summary:* Change STEP exporter to use `AP214 IS` by default. The default value of `write.step.schema` parameter has been changed to `AP214IS` as this schema is the current standard for writing STEP files.
25699	*Summary:* Crash while importing STEP file. Try-catch block has been added in `StepToTopoDS_TranslateEdgeLoop::Init()` to catch `Standard_ConstructionError` during the Curve conversion. Bspline curve degree is checked to prevent construction error when converting edges to `Bspline_Curve`.
25740	*Summary:* `VrmlData_Scene::WriteArrIndex()` writes extra point indices. The writing of extra point indices has been fixed in method `VrmlData_Scene::WriteArrIndex`.
25747	*Summary:* Bad IGES file after import and export with different units. The problems with scaling the radiuses of ellipse and hyperbola curves, cylinders, cones, spheres and toruses as well as scaling plane surfaces have been fixed in `GeomToIGES` package.
25816	*Summary:* IGES export – edges within compound are lost if BREP mode (IGES 5.3) is used. Writing wires, edges, and vertices inside a compound has been enabled in IGES in BREP mode. An IGES group is created for a compound even if it contains a single shape.

25843	*Summary:* Wire containing degenerated edge is not written to IGES / STEP. Check for edges with null 2D and 3D curves has been added in `TopoDSToStep_WireframeBuilder`. The problem with wires having several invalid edges and only one valid edge, which is not the last, has been fixed in `BRepToIGES_BRWire ::TransferWire`.
25910	*Summary:* The material with 0-density causes errors during writing STEP files. The method `STEPCAFControl_Writer::WriteMaterials` now properly processes 0-density materials and creates STEP density structures if and only if the density is >0, however, already mapped structures are preserved.
25912	*Summary:* Exception while reading STEP files with GD&T. Missing "`break`" statements have been added in method `RWStepAP214_GeneralModule::FillSharedCase`.
26138	*Summary:* Problems with writing periodic Bsplines into IGES. The algorithm writing periodic Bsplines into IGES has been modified in methods `GeomToIGES_GeomSurface::TransferSurface` and `BRepToIGES_BRWire ::TransferEdge.` Now Bspline surfaces are not converted into rational ones. P-curves are shifted to periodic Bspline surfaces and segments are cut from them.

Draw

22785	*Summary:* Add possibility to remove a text drawn by the command `vdrawtext` New public class `AIS_TextLabel` has been implemented to display simple text labels instead of private `MyTextClass`. It allows to easily clear labels from the Viewer. A reliable replacement for `VDisplayAISObject()` with no viewer update flag has been added in `ViewerTest::Display()`. The command `vdrawtext` now uses the new `AIS_TextLabel` class. Additionally: ▪ Parameter name and value syntax are used instead of a strict list of mandatory arguments. ▪ A redundant argument `isMultiByte` has been dropped. ▪ New argument –`noupdate` allows skipping the Viewer update.
25009	*Summary:* Incorrect handling of comma and period keys pressed in a 3D view. The unexpected effect of pressing comma and period keys in a Draw 3D view has been corrected.

25430	*Summary:* Dimensions demo fails. The following improvements have been introduced in Dimensions demo: ▪ `vdimparam` help string has been corrected; ▪ section model file has been added to prevent changing of subshapes order after `explode` call.
25697	*Summary:* Section obtained after command `bsection` in Test Harness is incorrect. The interface of command `bopcurves` has been extended. Now it is possible to obtain: ▪ only 3D-curves; ▪ 3D-curves and 2D-curves on one of intersected surfaces (the surface can be selected); ▪ 3D-curves and 2D-curves on every intersected surface.
25700	*Summary:* Ensure uniform control of the Boolean operations algorithm at the level of DRAW application. Parameter `[-s]` providing the sequential mode of computations has been removed from commands `bopcheck, bfillds, bbuild` and `bbop`. The mode of computations is now set by the command: `brunparallel [0/1]`, where 1 sets the parallel mode of computations and 0 sets the sequential mode of computations.
25801	*Summary:* Some commands in `BOPTest` package show the execution time without `-t` key. Unnecessary output of the execution time has been removed in commands `bopcheck, bfillds, bbuild` and `bbop`.
25825	*Summary:* Drop command `tovrml`. The command `tovrml` duplicating the functionality of `writevrml` has been removed.
25928	*Summary:* Tool for comparing curves. A set of commands has been implemented to debug intersection curves and already created projected curves. ▪ `xdistcc` checks distance between two 3d curves; ▪ `xdistcc2ds` checks distance between 3d curve and curve on surface (projected curve); ▪ `xdistc2dc2dss` checks distance between two curve on surface (projected curves). These commands are print 3d distance between input objects built on even grid. It is supposed that curves have the same parameterization.
25959	*Summary:* Do not turn on Z-clipping implicitly by mouse ring Possibility to switch the clipping planes implicitly by Ctrl + mouse ring has been disabled to avoid confusions.
26035	*Summary:* Length dimension can't be built between two vertices in Draw The command `vertex` has been corrected to properly draw the length dimension for the following objects: two vertices, face-point, point-face, edge-vertex/point and vertex/point-edge.

Mesh

21593	*Summary:* Integration of API to control addition of vertices from the Face into mesher. Additional parameter `InternalVerticesMode` (`-int_vert_off` for `incmesh` Draw command), which controls if the internal vertices of a face can be added into mesh, has been introduced in `BrepMesh_IncrementalMesh`.
24923	*Summary:* `BRepMesh_CircleTool` produces bad circles The function `BRepMesh_CircleTool::MakeCircle` now calculates the radius of circumcircle as the maximum difference between its center and the vertices of the reference triangle.
25378	*Summary:* Triangulation for distorted surfaces takes a very long time using `BrepMesh_IncrementalMesh`. New parameter `MinSize` has been introduced to `BrepMesh` and `GCPnts_TangentialDeflection` to limit amplification of tessellated representations of edges and surface mesh in case of highly distorted surfaces. The length of remaining curve part is additionally checked in method `BrepMesh_FastDiscretFace::insertInternalVerticesBSpline()` using min size parameter instead of distance between two points, which allows avoiding large gaps if Bspline surfaces are highly distorted.
25469	*Summary:* `BrepMesh` corrupts triangulation of another not connected shape. The method `BrepMesh_IncrementalMesh::update` now avoids cleaning polygons that are created for different faces not connected with the current shape. The changes producing additional points for Bspline curves with C1 continuity have been reverted.
25503	*Summary:* `BrepMesh` – `IncrementalMesh` always re-meshes the shape even if the existing triangulation satisfies the given deflection. In method `BrepMesh_IncrementalMesh::update()`, extraction of all triangulations and polygons by index has been replaced by iteration over faces pertaining only to the given shape and sharing an edge passed as argument. The classes `BrepMesh_FaceChecker` and `BrepMesh_EdgeChecker` have been removed.
25519	*Summary:* `BrepMesh` can break mesh regularity for Bspline surfaces. The method `BrepMesh_FastDiscretFace::insertInternalVerticesBSpline` now properly computes intervals to produce regular grid for Bspline surfaces and adds new internal points according to calculated values.
25547	*Summary:* Static class methods are not exported in `BrepMesh_GeomTool`. Static methods of `BrepMesh_GeomTool` class have become available from outside of `BrepMesh` package.

25612	*Summary:* Introduce the possibility to disable adaptive reconfiguration of triangles in `BrepMesh`. New flag `ControlSurfaceDeflection` (`-surf_def_off` for `incmesh` Draw command) enables or disables checking of mesh deflection from surface. This modifier prevents automatic rebuilding of the final mesh, performed to achieve a more precise result, however, its result can be worse than the initial one in case of Bspline surfaces.
25806	*Summary:* Stack overflow during meshing. The method `BrepMesh_Delaun::meshPolygon` has been fixed to prevent stack overflow during meshing of thin oblong polygons consisting of thousands of segments.
26028	*Summary:* Option for drawing `MeshVS_Mesh` as closed object. It has become possible to draw `MeshVS_Mesh` as a closed object, which enables such features as back face culling and capping.

Shape Healing

24881	*Summary:* Wrong status returned by `ShapeFix_Wire::FixGaps3d()` operation. `ShapeFix_Wire::FixGaps3d()` now checks the gap on adjacent points before trying to convert curves.
25013	*Summary:* `ShapeFix_Wire` tweaks for better results. The method `ShapeFix_Wire::FixEdgeCurves()` has been modified to recalculate the tolerance before edge cutting and to try to increase it before splitting in singularity during Pcurve adding.
25455	*Summary:* `Fixshape` works at the second attempt. New method `ShapeFix_Shape::FixVertexTolMode` has been added to check and fix tolerances of all vertices after performing all fixes (its default value is equal to -1). This avoids the situation when a vertex point belonging to a few faces is changed for the current face but the edges containing this vertex are not taken into account.
25520	*Summary:* Improvements in Shape Process and Shape Fix. The following improvements have been introduced in Shape Process and Shape Fix: ▪ The class `ShapeCustom_Modification` has been added as a base of `BsplineRestriction`, `ConvertToBSpline`, `ConvertToRevolution` and `DirectModification` from `ShapeCustom`; ▪ `ShapeCustom_Modification` has been modified to hold `BasicMsgRegistrator` from `ShapeExtend` where the descendant classes enumerated above can `SendMsg()`; ▪ Optional argument `ShapeExtend_BasicMsgRegistrator` has been added to `ShapeFix::SameParameter()`; ▪ `ShapeFix_FixSmallFace`, `ShapeFix_SplitCommonVertex` and `ShapeFix_Wireframe` have been modified to call `SendWarning()`; ▪ `ShapeFix_Root::SendMsg()` has been protected from accessing to NULL `myMsgReg`; ▪ All operators of `ShapeProcess_OperLibrary` have been instrumented with `ShapeExtend_BasicMsgRegistrator`; ▪ Optional argument `ShapeExtend_MsgRegistrator` has been added to `RecordModification()` and `RecordModification()`; ▪ The problem caused by the fact that `ShapeBuild_ReShape` can bind shapes with locations in its maps but `RecModif()` always asks for shapes without locations has been fixed. Now shapes with and without locations are checked (though only when recording messages); ▪ `ShapeUpgrade_ShapeDivide` has been modified to hold `BasicMsgRegistrator` from `ShapeExtend` where it `SendMsg()` during `Perform()`; ▪ Virtual methods `GetFaceMsg()`, `GetWireMsg()` and `GetEdgeMsg()` have been added to `ShapeUpgrade_ShapeDivide` to allow `ShapeUpgrade_ShapeConvertToBezier` to redefine them and to send appropriate messages.

25529 25604 25670 25743	*Summary:* `ShapeProcessAPI:` introduce `DropSmallSolids` operator New class `ShapeFix_FixSmallSolid`, which processes small solids, has been implemented. It provides the following methods: ▪ `Remove` allows removing small solids; ▪ `Merge` allows merging small solids to adjacent larger solids. Merging cannot be performed on shapes containing only small solids; ▪ `SetVolumeThreshold` sets or clears volume threshold for small solids; ▪ `SetWidthFactorThreshold` sets or clears width factor threshold for small solids; ▪ `SetFixMode()` sets the working mode for the operator: If `theMode = 0`, both `WidthFactorThreshold` and `VolumeThreshold` parameters are used; if `theMode = 1` only `WidthFactorThreshold` parameter is used; if `theMode = 2` only `VolumeThreshold` parameter is used.
25712	*Summary:* Non-deterministic behavior of `ShapeFix_Solid.` In class `ShapeFix_Solid` use of `TopTools_DataMapOfShapeListOfShape` has been replaced with `TopTools_IndexedDataMapOfShapeListOfShape` to avoid different results after the healing. All methods in class `BrepTools_ReShape` and subclass `ShapeBuild_ReShape` have become virtual. This feature helps to trace modifications stored and replayed in `ReShape`.
25823	*Summary:* Self-Intersecting wire translated from STEP file. The number of points used for computation of a bounding box of an edge has been increased in method `ShapeFix_ComposeShell::SplitByLine` to improve tolerance and the resulting bounding box.
26182	*Summary:* Calling `ShapeFix_FixSmallFace::RemoveSmallFaces()` always leads to stack overflow Unused methods `RemoveSmallFaces()` and `SplitFaces()` have been removed from class `ShapeFix_FixSmallFace`.

Configuration

24944	*Summary:* New custom-built Tcl causes of distribution problems. The option to link to MS run-time library statically has been added in the description of Tcl/Tk building from sources.
25146	*Summary:* Porting to Android OCCT has been ported to and now supports Android platform.
25498	*Summary:* Cmake refuses to build OCCT without any 3rd party libraries. Guards for empty Cmake variables have been added in `CmakeLists.txt`.
25745	*Summary:* Porting to iOS OCCT has been ported to and now supports iOS platform.
25791	*Summary:* Impossible to generate projects for Foundation Classes without `TclTk/Freetype` using Cmake. Redundant variables have been unset in `CmakeLists`.
25850	*Summary:* Installation procedure fails to find `*.pdb` files in debug mode in case of projects generated using Cmake. The path to `.pdb` files has been corrected in `osutils.tcl`.

Samples

25490	*Summary:* Error at the start of QT OCCT sample. Separate `msvc.bat` files have been created for each Qt sample to avoid errors.
25570	*Summary:* New Tcl sample scripts created for CAD Assistant. New sample scripts have been created in frame of the development of CAD Assistant for Android: ▪ `cpu.tcl` creates a colored model of Intel i-4790 CPU; ▪ `Penrose.tcl` creates a Penrose triangle made of boxes resembling the ones from OCC logo; ▪ `pencil.tcl` creates a colored pencil model; ▪ `snowflake.tcl` creates a 2d snowflake drawing.
26145	*Summary:* Geometry Sample crashes. The unacceptable usage of quantity coefficient has been fixed in Geometry sample.

Coding

25246	*Summary:* Make methods `Intervals` and `NbIntervals` const in `Adaptor3d_Curve` and its descendants. Qualifier `const` has been added to functions `NbIntervals` and `Intervals` in the following classes: `Adaptor3d_IsoCurve`, `Adaptor3d_Hcurve`, `Adaptor3d_Curve`, `Adaptor3d_CurveOnSurface`, `BrepAdaptor_CompCurve`, `BrepAdaptor_Curve`, `ChFiDS_ElSpine`, `GCPnts_TangentialDeflection`, `GeomAdaptor_Curve`, `GeomFill_SnglrFunc`, `HLRBRep_Curve`, `HLRBRep_BCurveTool` and `ProjLib_ProjectOnPlane`.
25546	*Summary:* Remove unused methods and classes from package `Aspect`. The following unused items have been removed from package `Aspect`: ▪ Global methods `Aspect::ToCString()`, `Aspect::ValuesOfFOSP()` and `Aspect::Inverse()`. ▪ Classes `Aspect_Edge`, `Aspect_Array1OfEdge` and `Aspect_EdgeDefinitionError`. ▪ Enumerations `Aspect_TypeOfFont`, `Aspect_TypeOfText`, `Aspect_CardinalPoints`, `Aspect_TypeOfRenderingMode`, `Aspect_TypeOfColorSpace` and `Aspect_FormatOfSheetPaper`.
25561	*Summary:* OCCT cannot compile with `OCCT_DEBUG` flag. `Function_Value` from class `ProjLib_PolarFunction` has been restored with the signature used in `OCCT_DEBUG` block.
25575	*Summary:* Remove `V3d_Static.hxx`. The class `V3d_Static` has been removed.
25616	*Summary:* Avoid classes using new to allocate Instances but not defining a copy Constructor. The following classes have been protected against copying: `Select3D_PointData`, `BSB_T3Bits`, `IntPatch_InfoPD`, `LDOM_StringElem`, `BinomAllocator`, `ProjLib_OnSurface` and `Standard_MmgrFactory`.
25619	*Summary:* CAST analysis: Avoid classes with a non-empty destructor and not implementing both an assignment operator and a copy constructor The destructors have been removed from the classes `tsee_entity`, `Select3D_PointData`, `Standard_MmgrFactory`, `ProjLib_OnSurface`, `BinomAllocator`, `OSD_PerfMeter`, `StorageInfo`, `OpenGl_UnpackAlignmentSentry`, `IntPatch_InfoPD`, `TableauRejection`, `Draw_View`, `BOPTest_Session`, `BOPCol_MemBlock`, `BSB_T3Bitsand` and `Ncollection_Handle::Ptr`.

25621	*Summary:* CAST analysis: Avoid constructors not supplying an initial value for all non-static data members. The constructors of the following classes have been fixed to provide initialization of all non-static fields: `Adaptor2d_Line2d`, `Adaptor3d_IsoCurve`, `Adaptor3d_OffsetCurve`, `AdvApp2Var_ApproxAFunc2Var`, `AIS_Dimension`, `AIS_InteractiveContext`, `Aspect_DisplayConnection`, `BiTgte_CurveOnEdge`, `BiTgte_CurveOnVertex`, `BrepAdaptor_CompCurve`, `BrepMesh_Circle`, `BrepMesh_Delaun`, `BRepToIGES_BREntity`, `ChFi2d_AnaFilletAlgo`, `ChFi2d_ChamferAPI`, `ChFi2d_FilletAlgo.cxx`, `ChFi2d_FilletAlgo`, `Extrema_ExtPExtS`, `Font_FTFont`, `GccEnt_QualifiedCirc`, `Geom2dAdaptor_Curve`, `IGESData_IGESEntity`, `IGESData_DefSwitch`, `IGESToBRep_CurveAndSurface`, `LDOM_XmlReader`, `math_TrigonometricFunctionRoots`, `NCollection_ListNode`, `ProjLib_CompProjectedCurve`, `ProjLib_ComputeApproxOnPolarSurface` and `Select3D_Box2d`.
25622	*Summary:* CAST analysis: Avoid invocation of virtual Methods of the declared Class in a Constructor or Destructor. `Delete()` methods have been removed from the following classes: `Adaptor2d_Curve2d` `Adaptor3d_Curve`, `Adaptor3d_Surface`, `AppBlend_Approx`, `AppCont_Function`, `AppParCurves_MultiCurve`, `AppParCurves_MultiPoint`, `ApproxInt_SvSurfaces`, `BrepPrim_OneAxis`, `BrepSweep_NumLinearRegularSweep`, `BrepSweep_Translation`, `BrepSweep_Trsf`, `DBC_BaseArray`, `GeomFill_Profiler`, `HatchGen_PointOnHatching`, `math_BFGS`, `math_FunctionSet`, `math_FunctionSetRoot`, `math_FunctionWithDerivative`, `math_MultipleVarFunction`, `math_MultipleVarFunctionWithHessian`, `math_MultipleVarFunctionWithGradient`, `math_Powell`, `math_NewtonMinimum` `math_NewtonFunctionSetRoot` `math_BissecNewton` (a virtual destructor added) `math_FRPR`, `math_BrentMinimum` (a virtual destructor added), `OSD_Chronometer` and `ProjLib_Projector`.
25629	*Summary:* `AIS_InteractiveContext` – code clean up. The following changes have been introduced in class `AIS_InteractiveContext`: ▪ Confusing method `AIS_InteractiveContext::Clear()` doing the same as `::Remove()` has been deleted. ▪ Double viewer update on first display of presentation has been fixed in method `AIS_InteractiveContext::Display()`. ▪ Retrieval of objects from local contexts has been `AIS_InteractiveContext::DisplayedObjects()`. ▪ The object is not bound twice in method `AIS_InteractiveContext::Load()`.
25684	*Summary:* Extend `Tcollection_ExtendedString` with method `IsEmpty()` The method `Tcollection_ExtendedString::IsEmpty` has been implemented to return `True` if the string contains no characters.
25734	*Summary:* GCC warnings in Android build. OCCT code has been revised to eliminate compiler warnings produced by GCC 4.7, mostly on unused or uninitialized variables.

25751	*Summary:* GCC warning `-Wunused-but-set-variable` in `gp_GTrsf2d.cxx` for Android build. Exception conditions of macros have been fixed in methods `gp_GTrsf2d::Trsf2d()` and `gp_GTrsf::SetForm()`.
25765	*Summary:* Coding rules – clean up code from obsolete macro checks. OCCT code has been revised to get rid of obsolete macro checks `GER61351`, `BUC60688`, `IMP160701`, `ALE70590`, `CTS17340`, `CSR577`, `PRO8619`, etc.
25770	*Summary:* Possible invalid memory access. Memory usage issues have been fixed in methods `Approx_SameParameter::Build` and `GeomInt_LineConstructor::TreatCircle`.
25790	*Summary:* Drop unimplemented method `ShallowCopy()` from `Tcollection_Hsequence.cdl`. Unimplemented method `ShallowCopy()` has been removed from `Tcollection_Hsequence.cdl`.
25849	*Summary:* Warnings on OCCT and PRODUCTS in 64-bit. Some warnings have been fixed in method `IVtkOCC_Shape::GetSubIds`.
26033	*Summary:* Get rid of `_Handle` classes OCCT has been revised to remove the explicit definitions of `handle` class in addition to the corresponding class itself. This is not necessary anymore as CDL syntax now imports classes inherited from `Standard_Transient`.
26167	*Summary:* Coding rules – eliminate `-Wlogical-not-parentheses` Clang warnings in `GeomToStep`. The package `GeomToStep` has been revised to avoid warnings about comparison statements that are logically correct but defined in a weird way.
26177	*Summary:* Coding rules – eliminate `–Wdeprecated-register` Clang warnings The warnings about deprecated `register` storage class specifier have been eliminated in classes `AdvApp2Var_MathBase`, `GeomLib` and `AdvApp2Var_SysBase`.

Documentation

23640	*Summary:* Documentation for local sewing with `BRepBuilderAPI_Sewing` is missing. An example of use of local sewing has been added in the documentation. Comments referring to the old location of sewing algorithm have been corrected.
25390	*Summary:* Redesign of the Technical Overview The Technical Overview has been redesigned for consistency with User Guides: duplicated information removed, images updated.
25527	*Summary:* Redundant references to `OpenCL.` Redundant references to `OpenCL` have been removed from the Overview.
25674 25702	*Summary:* Misprints in the documentation. Some misprints in the documentation have been fixed.
25723	*Summary:* `BRepOffsetAPI_ThruSections` fails for a case with open sections The description of method `BRepOffsetAPI_ThruSections::CheckCompatibility` has been updated.

WOK

25183	*Summary:* add `Standard_Override` to redefined methods. WOK has been improved to add `Standard_OVERRIDE` in generated C++ headers for class methods declared in CDL as “is redefined virtual”.
25585	*Summary:* Procedure of `CMakeLists` generation must be improved to make VTK product fully optional. Generation of `CMakeLists` for toolkits has been improved to make VTK product optional. If `USE_VTK` toggle is `OFF` during CMake configuration, VTK toolkits are not compiled.
25716	*Summary:* Some WOK projects are generated as executable instead of library. The configuration type has changed from application to dynamic library for the following WOK projects on Visual Studio 2010 or later: `wokcmd`, `wokdeliverysteps`, `wokdfltsteps`, `wokobjssteps`, `wokorbixsteps`, `woksteps`, `woktoolscmd` and `wokutilscmd`.
25744	*Summary:* Xcode projects generator – add extraction options. The following extraction options, which build generated xcode project, have been added: ▪ For Mac Os X: `wgenproj –target=xcd –static`, which builds generated xcode project; ▪ For Ios: `wgenproj –target=xcd –static –ios`.

Products

Advanced Samples

23640	*Summary:* Update QT product samples. Voxel Demo has been updated to correspond to the current state of OCCT.
25311	*Summary:* Add `BrepMesh` package in C# wrapper. C# wrapper has been added for class `BrepMesh_IncrementalMesh`.
25312	*Summary:* Improve C# wrapper to support SWIG 3.x. C# and Java wrappers have been improved to support SWIG 3.x. Parameter –DSWIG2_CSHARP can be used to include statements for `System` and `System.Runtime.InteropServices` for compatibility with SWIG 3.0.x;
25425	*Summary:* Contribution to C# Wrapper. Minor refactoring and improvements of C# and Java wrappers.
25543	*Summary:* Some product samples hang if the License is not found. Output information about missing license has been added in MFC and QT OMF samples.
25579	*Summary:* Small fixes in installation procedure for samples. Access to the resources has been provided to SSP Sample.
25977	*Summary:* Shape Healer cannot be compiled in 64 bit mode. The compilation of Shape Healer in 64 bit mode has been enabled.
25993	*Summary:* References to enumerations are wrapped wrong in SWIG Java. Java wrapper compilation has been provided with a workaround in `Aspect_DisplayConnection.hxx` to avoid name conflicts with OCCT methods.

Express Mesh

25434	*Summary:* The requested deflection is not satisfied. New tcl command `surface_deviation` has been introduced to check for deviation of the mesh from the real geometry. New functionality checking deviation of quad interior from face geometry has been implemented on level of quad tree division in case of Bspline surfaces.
25598	*Summary:* Express Mesh goes to infinite loop while checking of mutual intersection. `Minsize` parameter is now used instead of geometrical check in methods `QMShape_EnrichDiscrCurves::insertPoint` and `QMShape_Tessellator::DiscretiseEdge`.
25726	*Summary:* Express Mesh raises exception during meshing of face consisting of a single straight edge. The method `QMShape_Tessellator::ComputeQuadTreeOnFace` now checks if a discrete face has an outer wire consisting of at least one edge.
25727	*Summary:* Express Mesh produces bad triangles near face boundaries. The classification of nodes produced by `QuadTree` algorithm has been extended by additional check in 3d space, which is necessary in case of thin oblong faces.
25995	*Summary:* Express Mesh produces triangulation with free node. The following changes have been implemented in post processing of inner nodes of a quad tree in method `QMBgr_QuadTree::PostProcess()`: ▪ the shift of 3D point has been removed; ▪ the shift of 2D point has been decreased, `NextAfter` value of X coordinate is applied instead of the ratio of 2D sizes of the face.
26114	*Summary:* Discretizing a two-point wire that consists of two same edges ends up with exception. A condition for the wires with same edges has been added in the method `QMShape_Tessellator::DiscretiseWire`.

Mesh Framework

25707	*Summary:* Collect all connected mesh parts separately. The types `OMFBool_SplitElement` and `OMFBool_BooleanOperation` have been modified to collect the connected parts of any operation mesh separated by the split links. Each connected part is connected only by common links and collected separately from the other connected parts.
25761	*Summary:* Improve processing of coincident mesh parts. The following modifications have been implemented to improve processing of coincident mesh parts in Boolean operations: ▪ The method `OMFAlgo_MeshIntersect::Compute` now calculates the orientation of the loop for each of the elements. ▪ The contour mesher `OMFBool_MeshContour` now provides specific processing of several first fixed contours. ▪ The splitting of an intersected element into polygonal parts has been corrected in type `OMFBool_SplitElement` to provide presence of the polygonal parts corresponding to the other mesh elements intersecting this element by a nonzero area. ▪ New method `OMFAlgo_IntEF::SingleError` allows estimating the calculation error of a simple arithmetic operation on coordinates of a mesh; ▪ The method `OMFAlgo_IntPoint::Compute` now allows calculating the maximal shift of each mesh intersection point; ▪ The algorithm `OMFAlgo::PolygonNormal` calculates the normal of a polygon and an estimation of the calculation error of this normal based on the vertex tolerance. If the topological algorithm is not applicable, it is possible to calculate the orientation of the loop for each element using the geometrical algorithm `OMFAlgo_MeshIntersect::Compute`.
25826	*Summary:* Improve the consistency of the intersection of any element with any link in the element internal points. Type `OMFAlgo_IntEF` has been modified to improve consistency of intersection of any mesh link with any element of the other mesh. Now the intersection point is not calculated if the link intersects the element in more than one point or if both ends of the link are not located inside the element.
25873	*Summary:* Calculate only the bound links of any surface mesh A new optional parameter `theIsBoundOnly` of type `OMFControl_BoundaryEdges` allows calculating only bound links.
25874	*Summary:* Improve calculation of common intersection points of any two elements of different meshes The types `OMFAlgo_IntEF` and `OMFAlgo_MeshIntersect` have been modified to avoid non-common intersection points among the calculated common intersection points of two elements that belong to different meshes.'

25875	*Summary:* Improve the classification of the location of each section element relative to the other mesh New method `OMFBool_BooleanOperation::classifySection` classifies the location of each section element with a split link relative to the other mesh, basing on the location of this element relatively to the visible side of the other mesh. The bounding box of the other mesh is not used anymore for this purpose. The improved classification extends the BO to infinite closed meshes and intersected unclosed meshes.
25903	*Summary:* Convert the degenerated common parts of any intersected elements of the different meshes to polylines New method `OMFAlgo::PolygonToPolyline` converts any polygon to a polyline if the polygon is not degenerated to a point. It can be used by method `OMFAlgo_MeshIntersect::Compute` to convert the degenerated common parts of any intersected elements of different meshes to geometrically coincident polylines.
26090	*Summary:* Make the removal of unclosed contours in the intersection of the meshes optional The types `OMFBool_ErrorStatus` and `OMFBool_BooleanOperation` now can optionally check for unclosed contours in mesh intersection and remove them. Draw commands `Mfmeshcommon`, `Mfmeshfuse` and `Mfmeshcut` now output a message about presence of such contours if this option is used.
26092	*Summary:* Extend Draw command `Mfhidesel` to output the element identifiers Draw command `MFhidesel` has been extended to output the identifiers of hidden elements.
26093	*Summary:* Extend Draw command `MFfindnode` to the full precision The Draw command `MFfindnode` now can output the node position in full precision. Its behavior is managed by the parameter, which can be equal to 0 (short) or 1 (full). This feature is useful to access the precise position of a node.
26094	*Summary:* Create Draw tests for a mix of surface and solid objects Draw commands `MFmeshcommons` and `MFmeshcuts` have been added for testing of a mix of surface and solid objects.
26125	*Summary:* Measurement of the area and the bound length of a surface mesh Methods `Area` and `BoundLength`, which measure the area and the bound length of a surface mesh, have been implemented in type `OMFControl_MeshCharacteristics`. These methods are available in Draw using commands `MFmesharea` and `MFboundlength`.

26129	*Summary:* Classify all outer points for any triangle as such The algorithm `ClassifyPointOnFace` of type `OMFAlgo` now distinguishes between inner and outer points located on the straight line defined by a triangle side.
26137	*Summary:* Create an algorithm to calculate the surface mesh closeness The algorithm `IsClosed` has been implemented in type `OMFControl_MeshCharacteristics` to calculate the surface mesh closeness.
26140	*Summary:* Create a defined Boolean operation New operation `OMFBool_DEFINED` allows defining the mesh parts included in the result and their orientation. The corresponding command `MFbo` has been added in Draw. The mesh parts and their orientation in the result are defined by four integer values, which are interpreted according to type `TRelativeLocation` of type `OMFBool_BooleanOperation`.
26141 26175	*Summary:* Revise all Draw tests with status `BAD` The Draw tests for Mesh Framework Kernel have been revised and improved.

Surfaces from Scattered Points

26024	*Summary:* Too coarse shaded representation of the approximated surface (regression). The overlapping field has been removed from `SCATexturedShape.hxx`.

Collision Detection

25567	*Summary:* Bug and exception in the `ColDet` sample. The algorithm of collision detection has been fixed to avoid using in one thread the objects that have been deleted in another thread.

DXF/ACIS SAT Import / Export

23553 24102 25899 25986 25988	*Summary:* Fixes for ACIS Entity Reader and DXF Reader. DXF Import interface can now read the recent DXF versions, with ACIS data encoded in binary form. ACIS SAT interface can now read SAB (Standard ACIS Binary) files. Other features ▪ It has become possible to read `AcDsRecord` and `cyl_spl_sur`; ▪ `ReadHeader` and `ReadRecord` functions have been introduced; ▪ Support of `SURFACE` and `PLANESURFACE` has been added; ▪ Reading of version 21200, `SkinSplSur`, `OffSplSur` and `LoftSplSur` has been corrected.

New features

Shape Proximity Detector

The algorithm that computes intersections by generating tessellation (triangulation) of the source shapes and detecting overlapping of resulting meshes has been implemented to quickly detect intersecting pairs of subshapes.

In contrast to Boolean Operations, Partition and Self-intersection algorithms that compute topological intersections, new algorithm is based on mesh intersections and is expected to work much faster. However, the quality of the result depends on the quality of tessellation.

In addition, the result of this operation cannot be represented as a topological shape. A list of sub-shapes from each shape that localize the intersection is returned instead. This result might be helpful to the user for further analysis of the initial shapes.

By default the tolerance of intersection has negative value, which allows detecting intersections. A non-zero positive value allows detecting small gaps between shapes with a minor loss of the algorithm performance (as it is necessary to compute and take into account bounding box for each elementary triangle of the meshes).

Fuzzy Boolean Operations

Fuzzy Boolean operation is the option of Basic Operations to use additional tolerance. This option allows handling cases of touching and nearly coincident arguments.

The Fuzzy option is useful on the shapes with embedding or gaps between the entities of these shapes, which are not covered by the tolerance values of these entities. Such shapes can be the result of modeling mistakes, or translating process, or import from other systems with loss of precision, or errors in some algorithms.

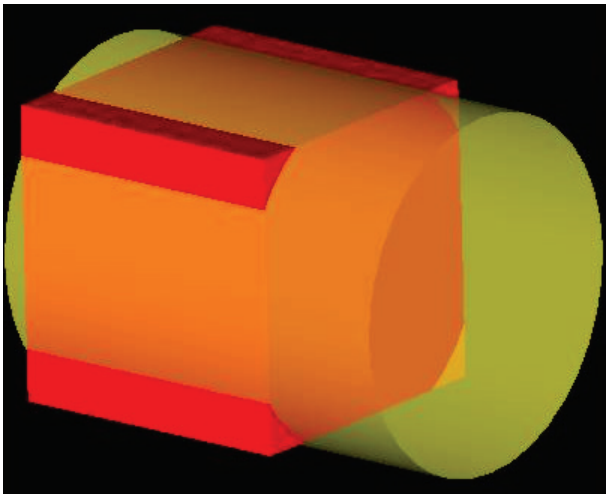
Most likely, the Basic Operations will give unsatisfactory results on such models: the result may contain unexpected and unwanted small entities, faulty entities (in terms of `BRepCheck_Analyzer`), or there can be no result at all.

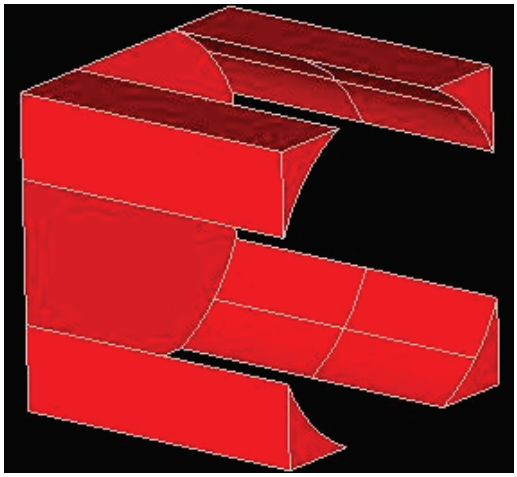
The Fuzzy option allows getting the expected result - it is only necessary to define the appropriate value of fuzzy tolerance for the operation. To define that value it is necessary to measure the value of the gap (or the value of embedding depth) between the entities of the models, slightly increase it to make the shifted entities coincident in terms of their tolerance plus the additional one and pass it to the algorithm.
Fuzzy option is included in interface of Intersection Part (class `BOPAlgo_PaveFiller`) and application programming interface (class `BRepAlgoAPI_BooleanOperation`).

Consider the example of the cylinder (shown in yellow and transparent) is subtracted from the box (shown in red). The cylinder is shifted by $5e^{-5}$ relatively to the box along its axis (the distance between rear faces of the box and cylinder is $5e^{-5}$).

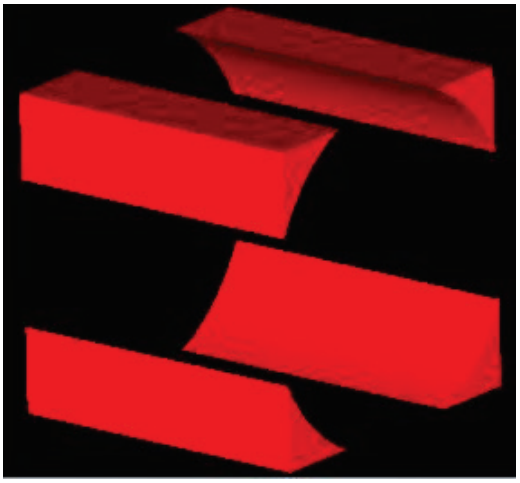
The following results are obtained using Basic Operations and the Fuzzy ones with the fuzzy value $5e^{-5}$

As it can be seen, Fuzzy option allows eliminating a very thin part of the result shape produced by Basic algorithm due to misalignment of rear faces of the box and the cylinder.





Result obtained with Basic Operations



Result obtained with Fuzzy Option

Boolean Operations with Multiple Arguments

Previously the Boolean Operator of Open CASCADE allowed processing of two and only two argument shapes: Object and Tool.

Now the Boolean Operator is able to process two groups of arguments with an arbitrary number of shapes, possibly having intersections with each other.

So, the updated API takes on input a list of shapes on each side, Object and Tool provided that each shape from the list meets usual requirements.

In the image below 30 tool shapes (shown in silver) are cut from the Object shape (shown in gold)

See the result of the operation:

The class `BRepAlgoAPI_BooleanOperation` contains the corresponding API. The methods `SetArguments(…)` and `SetTools(…)` provide the possibility to pass multiple arguments.

This feature gives the following benefits:

- Possibility to perform the operation in a single pass;
- Simplification of the caller procedure;
- Performance improvement due to usage of a common data structure;
- Performance improvement for the cases with a little number of actual intersections.

Porting to version 6.9.0

Porting of user applications from the previous 6.8.0 OCCT version to 6.9.0 requires the following issues to be taken into account:

Changes in Selection

In OCCT 6.9.0 selection mechanism of 3D Viewer has been redesigned to use 3-level BVH tree traverse directly in 3D space instead of projection onto 2D screen space (updated on each rotation). This architectural redesign may require appropriate changes at application level in case if custom Interactive Objects are used.

Standard selection

Usage of standard OCCT selection entities would require only minor updates.

Custom Interactive Objects should implement new virtual method `SelectMgr_SelectableObject::BoundingBox()`.

Now the method `SelectMgr_Selection::Sensitive()` does not return `SelectBasics_SensitiveEntity`. It returns an instance of `SelectMgr_SensitiveEntity`, which belongs to a different class hierarchy (thus `DownCast()` will fail). To access base sensitive it is necessary to use method `SelectMgr_SensitiveEntity::BaseSensitive()`. For example:

```
Handle(SelectMgr_Selection) aSelection = anInteractiveObject->Selection
(aMode);
for (aSelection->Init(); aSelection->More(); aSelection->Next())
{
   Handle(SelectBasics_SensitiveEntity) anEntity =
      aSelection->Sensitive()->BaseSensitive();
}
```

Custom sensitive entities

Custom sensitive entities require more complex changes, since the selection algorithm has been redesigned and requires different output from the entities.

The method `SelectBasics_SensitiveEntity::Matches()` of the base class should be overridden following the new signature:

`Standard_Boolean Matches (SelectBasics_SelectingVolumeManager& theMgr, SelectBasics_PickResult& thePickResult)`, where `theMgr` contains information about the currently selected frustum or set of frustums (see `SelectMgr_RectangularFrustum`, `SelectMgr_TrangularFrustum`, `SelectMgr_TriangularFrustumSet`) and `SelectBasics_PickResult` is an output parameter, containing information about the depth of the detected entity and distance to its center of geometry.

In the overridden method it is necessary to implement an algorithm of overlap and inclusion detection (the active mode is returned by `theMgr.IsOverlapAllowed()`) with triangular and rectangular frustums.

The depth and distance to the center of geometry must be calculated for the 3D projection of user-picked screen point in the world space. You may use already implemented overlap and inclusion detection methods for different primitives from `SelectMgr_RectangularFrustum` and `SelectMgr_TriangularFrustum`, including triangle, point, axis-aligned box, line segment and planar polygon.

Here is an example of overlap/inclusion test for a box:

```
if (!theMgr.IsOverlapAllowed()) // check for inclusion
{
  Standard_Boolean isInside = Standard_True;
  return theMgr.Overlaps (myBox.CornerMin(), myBox.CornerMax(), &isInside) &&
isInside;
}

Standard_Real aDepth;
if (!theMgr.Overlaps (myBox, aDepth)) // check for overlap
{
  return Standard_False;
}

thePickResult =
      SelectBasics_PickResult        (aDepth,        theMgr.DistToGeometryCenter
(myCenter3d));
```

The interface of `SelectBasics_SensitiveEntity` now contains four new pure virtual functions that should be implemented by each custom sensitive:

- `::BoundingBox()` – returns a bounding box of the entity;
- `::Clear()` – clears up all the resources and memory allocated for complex sensitive entities;
- `::BVH()` – builds a BVH tree for complex sensitive entities, if it is needed;
- `::NbSubElements()` – returns atomic sub-entities of a complex sensitive entity, which will be used as primitives for BVH building. If the entity is simple and no BVH is required, this method returns 1.

Each sensitive entity now has its own tolerance, which can be overridden by method `SelectBasics_SensitiveEntity::SetSensitivityFactor()` called from constructor.

Changes in Adaptor3d_Curve class

All classes inheriting `Adaptor3d_Curve` (directly or indirectly) must be updated in application code to use new signature of methods `Intervals()` and `NbIntervals()`. Note that no compiler warning will be generated if this is not done.

Changes in V3d_View class

The methods `V3d_View::Convert` and `V3d_View::ConvertWithProj()` have ceased to return point on the active grid. It might be necessary to revise the code of your application so that `V3d_View::ConvertToGrid()` was called explicitly for the values returned by `V3d_View::Convert` to get analogous coordinates on the grid. The methods `V3d_View::Convert` and `V3d_View::ConvertWithProj` convert point into reference plane of the view corresponding to the intersection with the projection plane of the eye/view point vector.

Supported Platforms and Pre-requisites

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

The most up-to-date information on Supported Platforms and Pre-requisites is available at http://www.opencascade.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
Android Operating System	Android 4.0.2 and above
iOS Operating System	iOS 7
Minimum memory	512 MB, 1 GB recommended
Free disk space (complete installation)	650 MB of disk space, or 1,4 GB if installed with reference documentation
Graphic library	OpenGL 3.3+, OpenGL ES 2.0+
C++ *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.