O p e n   C A S C A D E   T e c h n o l o g y   &   P r o d u c t s

Open CASCADE Technology and Products ver. 6.5.3 Maintenance Release

Release Notes

Overview

Open CASCADE Technology and Products version 6.5.3 is a maintenance release, which includes more than **200** new features, improvements and bug fixes over maintenance release 6.5.2.

Version **6.5.3** 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 6.5.3.

Highlights

- Optimization of packages related to interactive selection in 3D view

- Redesign of TKOpenGl graphic driver

- Improved management of gradient and textured background in 3D view

- Display of objects in overlay groups

O p e n   C A S C A D E   T e c h n o l o g y   &   P r o d u c t s

Table of Contents

Open CASCADE Technology & Products

Modifications

Foundation Classes

18387	*Summary:* Macro conflict for M_PI_2 and M_SQRT1_2 with math.h Definitions in Standard_math.hxx have been enclosed with #ifndef /#endif to avoid conflict with definitions from math.h.
21212	*Summary:* Missing Hasher parameter in the current implementation of NCollection Maps The following modifications have been introduced to implement Hasher parameter in NCollection Maps. ▪ New method NCollection_BaseList::Iterator::Initialize() has been added to provide coherence with TCollection_ListIterator class API. ▪ Additional template parameter has been added to classes DataMap, DoubleMap, IndexedDataMap, IndexedMap and Map from NCollection package. This parameter allows using user-specific hashers in the maps of this package. The default parameter is provided for compatibility (see class NCollection_DefaultHasher).
22611	*Summary:* Memory leak in expression interpreter Method ExprIntrp::Parse() has been improved to correctly call function ExprIntrp_stop_string() after parsing a string.
22668	*Summary:* Performance improvement in UBTreeFiller Two improvements have been made in class NCollection_UBTreeFiller to optimize the performance: ▪ Allocator instance has become a parameter of the constructor; ▪ Reset() method has been added. It can be called when UBTree should not be filled but UBTreeFiller has already accumulated some data. Without this method the UBTree is automatically filled by the destructor of UBTreeFiller.
22698	*Summary:* Add OCC_VERSION_DEVELOPMENT definition in Standard_Version.hxx Macro variable OCC_VERSION_DEVELOPMENT has been added to check new bug fixes that change the API in OCCT-based applications (for correct compilation with both final and development version).
22730	*Summary:* Info about exception misses The message shown when the execution is aborted due to unhandled exception has been corrected in class Standard_ErrorHandler.
22815 22816	*Summary:* new[] and delete[] operators redefined to use OCC memory manager. The following modifications have been implemented in OCCT: ▪ Macro DEFINE_STANDARD_ALLOC has been added in Standard_DefineAlloc.hxx class. It overrides memory operators to use OCC memory manager.

22815 22816	▪ WOK generator has been corrected to use macro DEFINE_STANDARD_ALLOC in generated classes. This macro also replaces redefinition of memory operators. ▪ StandardCSFDB functions have been replaced by DEFINE_STANDARD_ALLOC macro in class Standard_Persistent_proto. ▪ Old redefinition of memory operators in OpenGl package has been replaced by macro DEFINE_STANDARD_ALLOC. Macro IMPLEMENT_MEMORY_OPERATORS has been removed. ▪ Macro DEFINE_NCOLLECTION_ALLOC has been added in NCollection_DefineAlloc to define new and delete with NCollection_Allocator. ▪ Macros DEFINE_STANDARD_ALLOC and DEFINE_NCOLLECTION_ALLOC have been added to nested class NCollection_BaseCollection::Iterator. Redefinition of memory operator has been removed from all derived classes.
22744	*Summary:* Global HashCode function for TCollection_ExtendedString Global function TCollection_ExtendedString::HashCode has been implemented to allow usage of UTF16 strings as keys in data maps.
22749	*Summary:* Segmentation fault in HashCode() of Standard_Transient HashCode function has been removed from Standard_Transient and Standard_Persistent and replaced with HashCode from Standard_HashCode.
22753	*Summary:* Fixed mismatch new/delete in Standard_Failure.cxx Standard_Failure class has been corrected to properly use delete[] operator instead of delete.
22773	*Summary:* Compiler warnings on 64-bit MSVC in NCollection_Vector.hxx The types associated with array size, number of elements and item number in the array have been modified in NCollection_BaseVector to avoid compilation warnings.
22774	*Summary:* Memory leak in OSD_FontMgr::InitFontDataBase() method Memory leak in OSD_FontMgr class has been eliminated by use of C++ classes (OSD_Environment, TCollection_HAsciiString) instead of plain C code.
22797	*Summary:* Returned type Handle(Standard_Type)& should be changed to const Handle(Standard_Type)&. Some classes from Standard package have been corrected so that their method *_Type_() could return const Handle, to correspond to the definition of other types in OCCT.
22885	*Summary:* else clause applies to the wrong if statement because of missing braces The order of braces has been corrected in some cases to apply else clause to the correct if statement.
22889	*Summary:* Thread-safety in Standard_GUID Static variables in Standard_GUID class have been replaced with local ones to avoid breaking thread-safety.

22904	*Summary:* Clean up `sccsid` variables Obsolete definitions of `sccsid` variables used earlier to tag file versions by SCCS have been removed from OCCT classes.
22921	*Summary*: Macros for convenient inclusion of run-time asserts New macros implemented in class `Standard_Assert` allow including run-time asserts in the code in a convenient manner: the behavior is the same in Debug and Release mode, the developer is notified in Debug mode if `Assert` fails and is able to make a check and specify action by a single statement.
22931	*Summary:* Wrong `delete` operator Classes `Message_Msg` and `Message_PrinterOStream` have been corrected to properly use `delete[]` operator instead of `delete`.
22932	*Summary:* Memory leak in `math_FunctionRoots.cxx` Potential memory leak has been fixed in class `math_FunctionRoots`.
22953	*Summary:* `strcat` expects null-terminated destination string The code related to getting Ethernet address on obsolete platforms has been removed from OCCT
22954	*Summary:* Variable not freed upon `realloc` failure Class `OSD_Environment` has been improved to correctly handle variables.
22958	*Summary:* Redundant assignment Redundant values assignment has been removed from `Quantity_Color`.
22306	*Summary:* Improvement to debug memory leaks and insufficient memory growths Class `OSD_MAllocHook` has been modified to properly manage the situation when the number of freed blocks is greater then the number of allocated blocks. Negative values of allocated sizes are reported.
23019	*Summary:* `OSD_Chronometer` fails to compile due to lack of `clock_gettime()` on Mac OS X System function `task_info()` is used in `OSD_Chronometer::GetThreadCPU()` to retrieve process info on Apple platforms.
23033	*Summary:* `Standard_MMgrOpt::Reallocate` behavior must be similar to "`realloc`" `Standard_MMgrOpt::Reallocate` has been modified to allocate new memory block if NULL pointer has been passed.
23036	*Summary:* Incorrect initialization of `Poly_MakeLoops::Link` Constructor of `Poly_MakeLoops::Link` method has been revised to set correct value to '`flags`' member field.

Open CASCADE Technology & Products

Modeling Data

22761	*Summary:* Exception in debug mode The check that a variable has been initialized before accessing the value has been added in method BRepTools::AddUVBounds.
22910	*Summary:* Unable to compute isoline for a NURBS surface GeomLib constructor has been improved to avoid negative weights during the creation of a B-Spline curve, which might raise an exception.
22935	*Summary:* Crash in BRepFill_Sweep.cxx Call to DBRep::Set() has been corrected in class BRepFill_Sweep to avoid possible crash.
23027	*Summary:* Move TopAbs out of TKG2d TopAbs package has been moved from TKG2d to TKG3d.

Modeling Algorithms

21778	*Summary:* IGESControl_Writer does not take into account parameter write.iges.brep.mode The parameter write.iges.brep.mode has been correctly implemented in class IGESControl_Writer.
22117 22558	*Summary:* Wrong calculation in gp_Dir::Mirror(const gp_Ax2&) Method gp_Dir::Mirror has been fixed to provide correct calculation results.
22201	*Summary:* Unclear description of BRepTools_WireExplorer usage A more clear description of the constructor has been provided in class BrepTools_WireExplorer.
22488	*Summary:* Typo in Geom2d_BsplineCurve::LocateU() Method Geom2d_BsplineCurve::LocateU() has been modified to correctly use value adjusted to period for periodic B-Splines.
22500 22765	*Summary:* Sewing produced invalid shape from the compound with tolerance 0.1 Class BRepBuilderAPI_Sewing has been modified to avoid sewing small (length less than the specified sewing tolerance) edges lying on open boundary if such edges do not have overlapped parts.

Open CASCADE Technology & Products

22550	*Summary:* Improvement of thread safety The following modifications have been implemented in OCCT to improve thread safety: ■ Static variables have been replaced by field members in `Plate_Plate` class. ■ Chronometers have been implemented in a special compile mode (instead of debug) in classes `ProjLib_CompProjectedCurve`, `Approx` and `BRepAlgo_NormalProjection`. ■ Shared resources in `GeomConvert_ApproxCurve` and `GeomConvert_ApproxSurface` have been protected with `mutex`. ■ Static variables have been made local in packages `BRepLib`, `BRepClass3d`, `IntCurvesFace`, `IntCurveSurface` and `Intf` and class `BRep_Tool`. ■ Static variables have been made class fields in packages `ProjLib` and `Approx`. ■ Use of local buffers and wrapping with arrays has been implemented in package `IntImp` to avoid heap allocation and adverse impact on performance. ■ `EvaluationFunction` class has been implemented in packages `BSplSLib`, `BSplCLib`, `GeomConvert`, `Geom2dConvert`, `GeomLib` and `GeomPlate`. The following modifications have been implemented in OCCT to facilitate porting on Linux and MacOS platforms: ■ `max()` and `min()` have been replaced by `Max()` and `Min()` in class `AdvApp2Var_ApproxF2var`. ■ Method `NCollection_BaseAllocator::CommonAllocator()` has ceased to use `Standard_Mutex::Sentry` because static `Standard_Mutex` may be uninitialized. ■ Macros `abs`, `min` and `max` from AdvApp2Var package have been renamed to `advapp_abs`, `advapp_min` and `advapp_max` to avoid name conflicts.
22567	*Summary:* Speed up `math_FunctionSetRoot` Update of function F (`math_FunctionSetWithDerivatives`) by the last solution value has been added before all calls of method `GetStateNumber` in class `math_FunctionSetRoot`.
22623	*Summary*: Uninitialized variables in `HLRBRep_Curve::UpdateMinMax` in debug mode Packages `Contap`, `HLRAlgo` and `HLRBRep` have been revised to properly initialize local variables in debug mode.
22627	*Summary*: Change OCCT memory management defaults The default settings of OCCT memory management have been changed: ■ The default value of MMGT_OPT has been changed to 0 to use `Standard_MMgrRaw` wrapper over standard memory allocators; ■ The default value of MMGT_REENTRANT has been changed to 1 to make `Standard_MMgrOpt` thread-safe by default.
22690 22789	*Summary:* Exception in Encode regularity if an edge has no p-curve Methods from `Geom2dAdaptor_Curve`, `GeomAdaptor_Curve` and `GeomAdaptor_Surface` have been protected against access violation exception.
22694	*Summary*: Wrong result obtained after `GeomLib::ExtendSurfByLength` The flags for rational surface in u and v in `Geom_BSplineSurface` class have been set in the correct order to avoid deviation in extrapolating rational surfaces.

22717	*Summary:* Exception during sewing
	
	The following modifications have been implemented to improve sewing procedure:
	   ▪ `BrepLib::BuildCurve3d` has been improved to return `false` if the computation of a 3d curve has failed or if a 3d or planar curve has not been obtained on edge).
	   ▪ `FixConnected()` has been replaced by `FixConnected(num,preci)` in `ShapeFix_Wire.cxx`.

22727	*Summary:* Fillet produces no result Method `ChFi3d_Builder::CompleteData` has been modified to call the construction of `BrepBlend_AppSurface` with two additional arguments: `Degmax` and `Segmax`.
22735	*Summary:* Data races in BRepMesh working in parallel mode Processing of compounds has been corrected in BRepMesh to avoid data races in case of shared sub-shapes.
22736	*Summary:* Incorrect Transformation Class `gp_Trsf2d` has been modified to correctly implement transformation algorithm.
22747	*Summary:* Progress indicator in sewing algorithm Progress indicator has been implemented in sewing algorithm (class `BRepBuilderAPI_Sewing`).
22758	*Summary:* Problem in `BRepExtrema_DistShapeShape` Method `BRepExtrema_DistanceSS::Perform` has been modified to correctly assign solution points to the corresponding shapes.

22769	*Summary:* Sewing algorithm optimization
	
	The following classes have been added to optimize sewing algorithm performance:
	   ▪ Class `BRepBuilderAPI_VertexInspector` has been implemented to define the `Inspector` interface for `CellFilter` algorithm, working with gp_XYZ points in 3d space. This class can be used to search coincidence points with a certain tolerance.
	   ▪ Class `BRepBuilderAPI_BndBoxTreeSelector` has been implemented to select overlapping boxes stored in `NCollection::UBTree`. This class contains methods to maintain selection condition and to retrieve objects after search.

22771	*Summary:* Extra vertex produced by Boolean section in case of closed intersection curves New method `BOPTools_PaveFiller::PutClosingPaveOnCurve (BOPTools_Curve& aBC, BOPTools_SSInterference& aFFi)`, where <aBC> is the intersection curve and <aFFi> is Face/Face interference, has been implemented to provide pave sets for intersection curves that are closed 3D-curves.
22781	*Summary:* Regression in `GCPnts_Tangential Deflection` Improvements have been introduced in class `GCPnts_TangentialDeflection` to obtain polylines closer to the original curves.

22783	*Summary:* Improvement of `BRepOffsetAPI_MakeFilling` `BRepOffsetAPI_MakeFilling` algorithm has been enabled to store old boundary edges and to create new boundary edges containing p-curves from the old ones. Method `BRepOffsetAPI_MakeFilling::Generated(aShape)` has been redefined to get new edges from old ones.
22786	*Summary:* 64bit issue in `AdvApp2Var` Casts from pointers to long integers have been fixed in `AdvApp2Var` package to avoid potential crashes on 64-bits platforms.
22790	*Summary:* Boolean operation `Fuse` fails New method `IntTools_EdgeEdge::IsSameCurves()` returns `Standard_True` if the 3D curves (B-Splines) obtained by `copy` are the same. The method improves Boolean operations by avoiding unnecessary deep computations.
22798 22809	*Summary:* `BRepBuilderAPI_GTransform` produces invalid result on a sphere The following modifications have been implemented to improve the results of scaling with different scale factors along axes. ▪ Conversion algorithm in `BRepTools_NurbsConvertModification` has been changed to work with greater precision. ▪ Processing of p-curves in `ProjLib_ComputeApproxOnPolarSurface` has been changed to work with greater precision. ▪ New Draw command `scalexyz` has been added to test results of scaling.
22801	*Summary:* Boolean operations algorithm fails on two shapes ▪ Check of array bounds has been added in method `ApproxInt_MultiLine::MakeMLBetween` to avoid instability. ▪ The tolerance value has been set to default `Precision::PConfusion()` in `IntTools_Context.cxx`.
22802	*Summary:* Excess allocated memory is not released New method `BOPTColStd_CArray1::Purge()` has been implemented to release the memory that is allocated but unused.
22804 22846	*Summary:* Regression on a test case A Boolean flag defining that the minimal tolerance is specified has been added in class `BRepTest_SurfaceCommands`.
22814	*Summary:* Boolean operation Section fails The loop checking the coincidence of points on a walking line has been changed in class `IntTools_FaceFace`.
22823	*Summary:* Exception in `thickshell` operation `BrepOffset_MakeOffset` class has been corrected to correctly apply `thickshell` command.

22828	*Summary:* Fuse operation crashes on simple shapes with conical surfaces The algorithm processing cases with coinciding cone apexes of has been fixed in method `IntAna_QuadQuadGeo::Perform()`.	
22833	*Summary:* Draw command `checkshape` gives error "Self intersection wire" on a shape. `IntCurve_IntConicConic` has been improved to correctly find the intersection point of two lines.	
22851	*Summary:* No intersection curve between two surfaces Method `Adaptor3d_TopolTool::BsplSamplePnts()` has been enabled to process cases when the number of computed points on a B-Spline surface is less than the given minimal value.	
22864	*Summary:* Algorithm `BiTgte_Blend` obtains instable results Data structures of types `Map` and `DataMap` have been replaced by corresponding data structures of types `IndexedMap` and `IndexedDataMap` in some classes of packages `BrepOffset` and `BiTgte` to provide stable work of the algorithm.	
22881	*Summary:* Sewing does not sew edge to its copy Class `BRepBuilderAPI_Sewing` has been improved to correctly connect faces.	
22887	*Summary:* Make class `Intf_InterferencePolygon2d` thread-safe. The following modifications have been introduced to improve thread safety of class `Intf_InterferencePolygon2d`: ▪ Class `Intf_InterferencePolygon2d` has ceased to be generic (template). Specific types of polygons can be processed using virtual methods of `Intf_Polygon2d` class that is a root class for all polygons that can be handled by `Intf_InterferencePolygon2d`. ▪ `Intf_ToolPolygon` class has been removed, its functionality is now provided by the root class `Intf_Polygon2d`. ▪ Each specific polygon class inherits `Intf_Polygon2d` and implements special virtual methods that are called by `Intf_InterferencePolygon2d`.	
22907	*Summary:* Function `distmini` produces wrong result Improvements have been introduced in `Extrema_FuncExtPC` case to properly process the case when two knots of a BSpline curve are the same.	
22923	*Summary:* Command `intersect` throws an exception The following modifications have been implemented to improve partition procedure: ▪ New field `myDegenerated` has been added in `IntPolyh_Point` to indicate if the point belongs to a degenerated surface boundary or not. ▪ New methods `IntPolyh_Point::SetDegenerated()` and `IntPolyh_Point::Standard_Boolean Degenerated()` correspondingly set and return the flag that indicates whether the point belongs to a degenerated surface boundary or not. ▪ Processing of triangles having a degenerated edge has been improved in class `IntPolyh_Triangle`.	

22923	• New method `IntPolyh_MaillageAffinage::FillArrayOfPnt` adds to all points of the surface grid the flag that indicates if the point belongs to a degenerated surface boundary or not.
22936	*Summary:* Memory leak in `inc\MAT_Mat.gxx` C pointers have been replaced by `TColStd_Array1OfInteger` in class `MAT_Mat` to avoid possible memory leak.
22937	*Summary:* Wrong `delete` operator Dynamic variable allocation has been replaced by static in classes `GeomFill_EvolvedSection` and `GeomFill_UniformSection`.
22938	*Summary:* Memory leak in `HLRBRep_InternalAlgo.cxx` C pointers have been replaced by array classes from `TcolStd` package in class `HLRBRep_InternalAlgo` to avoid possible memory leak.
22944	*Summary:* `TopOpeBRepTool_traceSTATE.cxx` Redundant declaration of `Standard_EXPORT` has been removed from class `TopOpeBRepTool_traceSTATE`.
22946	*Summary:* `BRepFeat_SplitShape` crashes on splitting a face by two edges The following new methods have been introduced to improve shape splitting: • `BRepFeat_SplitShape::Add(aCompound, aFace)` adds a compound on face. The compound must consist of edges lying on the face. If the edges are geometrically connected, they must be connected topologically, i.e. they must share common vertices. • `LocOpe_ProjectedWires::IsFaceWithSection(aFace)` and `LocOpe_WiresOnShapeIsFaceWithSection(aFace)` determine if a face should be split by the section or not. • `LocOpe_WiresOnShape::Bind(aCompound, aFace)` adds the edges of <aCompound> and <aFace> to the inner data structure of class `LocOpe_WiresOnShape`, moreover, it adds <aFace> to the list of faces that should be split by a section. • `LocOpe_ SplitShape::Add(aListOfWires, aFace)` allows adding a section on a face.
22946	Additionally: • `LocOpe_SplitShape::BindAll()` has been modified to detect internal intersections of new edges with boundary edges. • Command `splitshape` from `BRepTest_FeatureCommands` has been modified to add a compound of wires or edges on a face.
22967	*Summary:* Boolean operations between two cylinders with orthogonal axis generate a shape with big vertex tolerance The following modifications have been implemented to improve Boolean operations: • Methods `SetContext()` and `Handle_IntTools_Context& Context();` have been implemented in several classes of `IntTools` package to correspondingly set and return intersection context. • The tolerance value of 3D intersection curve has been refined in static function `Standard_Real MaxSquareDistance()` from class `IntTools_FaceFace`.

22968	*Summary:* RebuildShape exchanges 2d-curves Method ShapeUpgrade_RemoveLocations::RebuildShape now avoids misplacing curves in BRep_TEdge in case of a seam reversed oriented TopoDS_Face.
22969	*Summary:* Confusion in BRepLib_MakeEdge Class BrepLib_MakeEdge has been corrected to use Precision::Pconfusion() instead of Precision::Confusion() on a curve.
22989	*Summary:* BSplCLib::Reparametrize() fails on near knots Method BSplCLib::Reparametrize has been corrected to properly shift knot values when two knots are too near to each other.
22990	*Summary:* Sphere splitting across the seam is incomplete Method LocOpe_SplitShape::ChoixUV(...) has been improved to calculate the distance between the last vertices on edges in 3D space, not in 2D as previously.
22998	*Summary:* Various exceptions in HLR algorithms Additional checks have been introduced in HLRBRep_Data to avoid exceptions.
23004	*Summary:* Boolean operation cut produces incorrect result The following new methods have been introduced in class BOP_FaceBuilder to improve Boolean operations algorithms: ▪ BOP_FaceBuilder::SetContext(const Handle(IntTools_Context)& aCtx) to set the object of intersection context <aCtx> required for the algorithm ▪ BOP_FaceBuilder::Context()const returns the object of intersection context ▪ BOP_FaceBuilder::PerformAreas(BOP_WireEdgeSet& aWES) provides classification of convexities and holes. The parameter <aWES> contains all necessary data for the method. New field Handle(IntTools_Context) myContext contains the object of intersection context
23004	The following methods have been removed from class BOP_FaceBuilder as obsolete: ▪ void BuildNewFaces() ; ▪ void SetManifoldFlag(const Standard_Boolean aMFlag) ; ▪ Standard_Boolean ManifoldFlag() const; ▪ Standard_Integer InitFace() ; ▪ Standard_Boolean MoreFace() const; ▪ void NextFace() ; ▪ Standard_Integer InitWire() ; ▪ Standard_Boolean MoreWire() const; ▪ void NextWire() ; ▪ Standard_Boolean IsOldWire() const; ▪ const TopoDS_Shape& OldWire() const; ▪ const TopoDS_Wire& Wire() const; ▪ void FindNextValidElement() ; ▪ Standard_Integer InitEdge() ; ▪ Standard_Boolean MoreEdge() const; ▪ void NextEdge() ; ▪ const TopoDS_Edge& Edge() const; ▪ void MakeLoops(BOP_WireEdgeSet& SS).

23008	*Summary:* Too many intersection vertices between line and cylinder Processing of intersections between line and cylinder has been added in method `IntTools_BeanFaceIntersector::FastComputeExactIntersection()`.
23031	*Summary:* Empty result for the extrema between a circle and a line The algorithm finding extreme distances between a straight line and a circle has been improved.
23043	*Summary:* Wrong results of `BRepExtrema_DistShapeShape` Method `Extrema_ExtElCS::Perform` has been improved to correctly find the distance between intersecting line and cylinder.
23051	*Summary:* Problem with `BRepExtrema_DistShapeShape` The problem with swap of returned solution points for `extrema` between face and vertex has been fixed in `BRepExtrema_DistanceSS`.
23060	*Summary:* Exception is raised during intersection of two edges The following modifications have been implemented to avoid possible exception: ▪ Method `BOPTools_PaveFiller::PerformEE()` has been protected against overwriting. ▪ Conditions defining a shrunk range have been changed in `BOPTools_PaveFiller::CorrectShrunkRanges` to provide the right order of boundaries.
23076	*Summary:* Empty result of intersection algorithm for a curve and a surface The condition determining it the surface should be trimmed or not has been changed in method `IntCurveSurface_Inter::DoTrim` to allow processing of surfaces with big 2D aspect ratio values.
23089	*Summary:* Wrong result of solid classifier algorithm Method `IntCurvesFace_Intersector::IntCurvesFace_Intersector` has been modified to restrict the usage of inner polyhedron for the surfaces that have big values of 2D aspect ratio.
23092	*Summary:* Error in `BRepProj_Projection` algorithm Method `TopOpeBRepTool_tol::FTOL_FaceTolerances` has been modified to calculate tolerances "myTol 1" and "myTol 2" are as a sum of tolerances of two faces (as in `IntTools_FaceFace`).

Open CASCADE Technology & Products

Open CASCADE Technology & Products

Visualization

Summary: Optimization of existing selection classes

`SelectBasics`, `Select3D`, `SelectMgr` and `StdSelect` packages for interactive selection in 3D view have been revised and optimized: unused methods have been removed and class hierarchy has improved.
More efficient memory use has been implemented as `double` (8 bytes) values have been replaced by `float` (4 bytes) wherever applicable, unused data fields have been removed, misplaced ones have been moved to the proper abstraction levels. The changes resulted in ~30% less memory occupied by data structures used for interactive selection on meshes (`MeshVS_Mesh` presentation).

12121
22039

The following code revisions and improvements have been implemented.
In `SelectBasics` package:
- Obsolete field `myusers` and related methods have been removed from `SelectBasics_EntityOwner` class; pure virtual methods working with locations have been added in this class to eliminate cyclic dependency `Select3D <-> SelectMgr`; methods related to priority have become `inline` for more efficiency.
- Field type of `mySFactor` has been changed to `ShortReal` in `SelectBasics_SensitiveEntity` class.
- Obsolete method `SelectBasics_SensitiveEntity::Size()` has been removed.

In `SelectMgr` package:
- Field "myLoc" has been removed from `SelectMgr_EntityOwner` as location management methods now rely on the corresponding methods of `SelectMgr_SelectableObject`.
- Empty re-implementation of `UpdateLocation(me: mutable; P : mutable Presentation from Prs3d)` has been removed from `SelectMgr_SelectableObject` and `AIS_ConnectedInteractive` classes.
- Obsolete field "mySize" and related methods have been removed from `SelectMgr_SortCriterion` class.
- `TopoDS_Shape` object has been moved from a generic class `SelectMgr_EntityOwner` to entity owner implementation `Select_BrepOwner`.
-
In `StdSelect` package:
- Fields "myShape" and "myFromDecomposition" and related methods have been moved from `SelectMgr_EntityOwner` to `StdSelect_BrepOwner` class;
- Obsolete method implementation `StdSelect_BrepSelectionTool::Load()` with "Users" argument has been removed from `StdSelect_BrepSelectionTool`.

In `Select3D` package:
- New class `Select3D_SensitivePoly` provides common base class for sensitive polylines and polygons. This class holds 3D points and computes projected 2D points and 2D bounding box and stores them in a compact form with `ShortReal` coordinates
- New class `Select3D_Box2d` provides a 2D bounding box with compact memory representation due to `ShortReal` co-ordinates.

12121 22039	- New class `Select3D_Pnt` provides a 3D point with `ShortReal` coordinates and automatic conversion to `gp_Pnt` and `gp_XYZ`. - New class `Select3D_Pnt2d` provides a 2D point with `ShortReal` coordinates and automatic conversion to `gp_Pnt2d` and `gp_XY`. - Macro `DtoF()` has been implemented in `Select3D_Macro` for safe conversion from `double` to `float` - Obsolete method `ComputeSize()` has been removed from all classes of `Select3D` package. - Methods `Project()`, `Areas()` and `Points3D()` as well as fields holding 3D and 2D geometry have been moved from classes `Select3D_SensitiveCircle`, `Select3D_SensitiveTriangle` and `Select3D_SensitiveFace` to the base class `Select3D_SensitivePoly`. - Method `Select3D_SensitiveEntity::SetLastDepth()` has been added to avoid direct modification of "mylastdepth" field by successors. - Methods `Select3D_SensitiveSegment::StartPoint()`, `Select3D_SensitiveSegment::EndPoint()` and `Select3D_SensitivePoint::Point()`, have been modified to return `gp_Pnt` object constructed on-the-fly. - Field "myLoc" has been removed from `Select3D_SensitiveEntity` class as location management methods now rely on the methods of the entity owner. - Fields "mylastsize" and "myhascomputed" have been removed from class `Select3D_SensitiveEntity`. - Fields "myX" and "myY" have been moved to `Select3D_SensitiveGroup` from `Select3D_SensitiveEntity` with type changed to `ShortReal`. - Field size of "myLastTol" has been changed to `ShortReal` in `Select3D_SensitiveGroup`. - Field types of "mypoint" and "myprojpt" have been changed in `Select3D_SensitivePoint`. - Types of fields holding 3D and 2D points have been changed in `Select3D_SensitiveSegment` class. In other packages: - Method `AIS_ConnectedInteractive::UpdateLocation(me: mutable; P: mutable Presentation from Prs3d)` has been removed. - Method `AIS_ConnectedShape::ComputeSelection` has been implemented. - Obsolete method `MeshVS_SensitivePolyhedron::ComputeSize()` has been removed. - In XSDRAWSTLVRML_DataSource class field types of "myElemNodes" and "myNodeCoords" have been changed from map to `Harray2` to optimize memory usage Unnecessary field "myElemCoords" has been removed and field "myElemNormals" has been added to calculate mesh face normals in this class.
12121 22039	New `Draw` commands: `vconnect` – creates and displays `AIS_ConnectedInteractive` object from the given object with a new location `vconnectsh` – creates and displays `AIS_ConnectedShape` from the given shape with a new location `vselmode` – sets the selection mode for the named object or for all displayed objects
17100	*Summary:* Buffer overflow vulnerability and loading TKOpenGl without environment variables on UNIX systems TKOpenGl graphics library now can be defined by `CSF_GraphicShr` environment variable; otherwise it is searched for in the system library path.

17100	The following changes have been implemented in the frame of this improvement: ▪ Possible buffer overflow has been corrected in method `Draw::Appli`. Now the length of the string, which defines path `/DrawResources/DrawDefaults`, can be more than 128 symbols. ▪ `Graphic3d_GraphicDevice` has been enabled to load `TKOpenGl` library without `CSF_GraphicShr` environment variables. If `CSF_GraphicShr` variable cannot be found, the library is searched for in system library paths. ▪ The declaration of method `Graphic3d_GraphicDevice::ShrIsDefined (Tcollection_AsciiString& aShr)` has changed to `Tcollection_AsciiString ShrEnvString()`. Now this method returns the string for loading a graphic driver from the environment path or a system library path.
22018	*Summary:* `Dmin` parameter left uninitialized The following modifications have been implemented in class `Select3D_SensitiveCircle`: ▪ Two new members: `myCenter3D` and `myCenter2D` have been added. `myCenter2D` is initialized as a projection of `myCenter3D` and is used in method `Matches()` to calculate parameter `Dmin`. ▪ New method `Select3D_SensitiveCircle::Project()` redefined from `Select3D_SensitivePoly` calls a default `Project()` and then projects `myCenter3D` on `myCenter2D`. ▪ New method `Select3D_SensitiveCircle::ComputeCenter3D()` initializes `myCenter3D` as the barycenter of points from `mypolyg3d`. ▪ `Select3D_SensitiveCircle` constructors initializing `myCenter3D` as the center of `TheCircle` or by calling `ComputeCenter3D()` have been added. ▪ If member `myFilledStatus` from `Select3D_SensitiveCircle::Matches` is `Standard_True`, `Dmin` is initialized as the distance between the mouse point and the center of the circle. ▪ Draw command vcircle has got one additional parameter `IsFilled` and draws a circle based on `AIS_Circle` if `IsFilled=0` or draws a filled circle based on `FilledCircle` if `IsFilled=1`.
22234 22819 22880 22955 22956 23001 23015 23035	*Summary:* Redesign of `OpenGl` driver OCCT `OpenGl` driver (`OpenGl_GraphicDriver` API class) has been redesigned to introduce C++ classes for all major entities, instead of the current C style (structures and separate functions). The code in `OpenGl` package has been revised, without touching the present high-level visualization API. The main purpose of this action was to facilitate maintenance of OCCT visualization engine and improve its robustness. Although the code changes were internal by their nature, the most important ones are mentioned below. This information might be helpful for the advanced OCCT users who are not afraid of getting inside `TKOpenGl`. The following new classes have been created to re-implement `OpenGl` package functionalities: ▪ `OpenGl_Context` class represents low-level GL context. On initialization it retrieves available GL extensions (currently `OpenGl_ArbVBO` and `OpenGl_ExtFBO`). This class replaces old mechanisms, such as `QueryExtension()` in "`OpenGl_Extension.hxx`" header.

	22234 22819 22880 22955 22956 23001 23015 23035	`OpenGl_Display` represents objects that control common properties of all contexts / windows (on UNIX it is bound to X Server connection). This class stores global (shared between GL contexts) primitives, such as markers and provides text-rendering methods. `OpenGl_Window` represents a window in GL context. This class stores native system-dependent handles and provides low-level functions to activate or swap front and back buffers. `OpenGl_Workspace -> OpenGl_Window` encapsulates the functionality that previously existed with WS suffix and maintains a set of "global" attributes (old `TsmSetAttri`, `TsmGetAttri`, etc). This class controls the current GL context state using `OpenGl_Aspect*` elements. It can be used as the container (or medium) of various statuses and attributes in method `Render()` of all primitive elements. `OpenGl_View` represents the part of the window where the graphic scene is represented. Now only one view is allowed per window. - Virtual method `OpenGl_Element::Render()` represents a new drawable interface. It is implemented by all drawable primitives: `OpenGl_Marker`, `OpenGl_MarkerSet`, `OpenGl_Polygon`, `OpenGl_PrimitiveArray` and `OpenGl_Text` as well as aspects: `OpenGl_AspectFace`, `OpenGl_AspectLine`, `OpenGl_AspectMarker` and `OpenGl_AspectText`. `OpenGl_PrimitiveArray` represents an array of drawable primitives. This is the only method to render geometry in the redesigned driver. `OpenGl_Group -> OpenGl_Element` represents a group of drawable primitives stored as sequence. This class corresponds to `Graphic3d_Group`. Notice that the group aspects have been placed as class members rather than just aspects in front of all primitives. Additional aspects that can be placed as elements within the group sequence are still supported. `OpenGl_Structure -> OpenGl_Element` represents a structure or a set of groups with own attributes and maps to `Graphic3d_Structure`. The following modifications have also been introduced: `OpenGl` has become a no-cdl package (however, its state was not changed to `'nocdlpack'` in UDLIST), so all classes are declared using C++ headers. - Most of memory allocation/deallocation mismatches have been fixed. - Most compiler warnings have been removed. - Compatibility workarounds for Windows 95 system have been removed. - Compatibility with OpenGL 1.0 standard has been canceled. OpenGL 1.1+ is required for compilation. - Double buffer emulation using `GLX Pixmap (UNIX)` has been removed. - All low-level static functions (`Tsm*`, `Txgl*`, `Tel*`, `call_*`) have been removed. `QuadrangleSet`, `Bezier`, `PolygonHoles`, `QuadrangleMesh`, `TriangleMesh` and `TriangleSet` primitive types have been removed from `OpenGl_GraphicDriver`, `Graphic3d_GraphicDriver` and `Graphic3d_Group`. - Comprehensive debug messages have been added for "vprintview" command
	22332	*Summary:* Fix for `Visual3d_Layer` (`linetype`, `linewidth` and `transparency`) The following modifications have been introduced to properly implement `linetype`, `linewidth` and `transparency` features : In `OpenGl` package Method `call_togl_set_line_attributes` that sets line attributes in layer mode has become independent on static flag variables; instead it tries to acquire the current `OpenGl` attribute flags.

22332	▪ Methods `call_togl_begin_layer2d` and `call_togl_end_layer2d`, which define layer beginning and end correspondingly, have been modified to keep `OpenGl` attribute flags unchanged. ▪ `TEL_LS_MAX` value is used for line style list creation in `TsmInitAttributes`. In `InterfaceGraphic` package ▪ The maximum count of default line styles has been defined as `TEL_LS_MAX` in `InterfaceGraphic.hxx`. In `ViewerTest` package: ▪ DRAW command "`vlayerline`" has been added to test `linewidth`, `linetype` and `transparency` of `Visual3d_Layer` class.
22368	*Summary:* `AIS_Trihedron` highlighted presentation is not updated when a trihedron is relocated The following modifications have been implemented to correct shapes selection in `Trihedron` algorithm: ▪ Active selection shapes have been updated with calls to methods `SetComponent`, `ComputeSelection` and `SetContext` from `AIS_Trihedron` class ▪ Method `AIS_Trihedron::SetLocation(const TopLoc_Location& aLoc)` has been overridden from the base class; Now the selection shapes are updated on `SetLocation`. ▪ Method `AIS_Trihedron::ComputeSelection` now adds selection shapes to the context and hides their display presentation. Once unselected, they will be displayed on highlight and selection only. ▪ `StdSelect_BrepOwner` class has been corrected: the presentation shape is nullified and recomputed on highlighting if the update flag is set, so the context can cancel highlighting of the old presentation shapes correctly. ▪ `SelectMgr_SelectableObject` now calls method `SelectMgr_EntityOwner::SetLocation` when its own `Location` is updated. ▪ `AIS_Trihedron::SetLocation(const TopLoc_Location& aLoc)` has been overridden from the base class `PrsMgr_PresentableObject`. ▪ New Draw command `vsetlocation : name x y z` allows changing AIS interactive object location through the interactive context with method `SetLocation()`.
22523	*Summary:* DRAWEXE randomly crashes during smooth navigation in 3D-Viewer Method `NIS_View::Select()` has been protected against too small movement vector `anXdir`.
22554	*Summary:* Application hangs on selection The following modifications have been implemented in the frame of `Selection` mechanism improvement: ▪ New class `Select3D_PointData` has been added for safe management of `Select3D_SensitivePoly` polygons of 3D and 2D points. ▪ Members `mypolyg3d`, `mypolyg2d` and `mynbpoints` have become obsolete and were removed from classes `Select3D_SensitiveCircle`, `Select3D_SensitiveCurve`, `Select3D_SensitiveFace`, `Select3D_SensitivePoly` and `Select3D_SensitiveTriangle`. ▪ Two new classes `Triangle` and `SegmentObject` based on `AIS_InteractiveObject` and two new Draw commands `vtriangle` and `vsegment` have been added in `ViewerTest_ObjectCommands.cxx` to test `Select3D_SensitiveTriangle` and `Select3D_SensitiveCurve` correspondingly.

22649	*Summary:* Export to EMF is not available EMF export patch to `gl2ps` library has been prepared. Corresponding modifications have been introduced in `Graphic3d`, `OpenGl` and `Visual3d` packages.
22683	*Summary:* Incorrect result of `Select3D_Projector::Project()` in case of perspective projection Obsolete and unused methods have been removed from `Select3D_Projector` class. `Projector` class description in documentation and Visualization User's Guide has been updated to make its usage easier.
22704	*Summary:* Define `TKOpenGl.dll` for OCCT visualization toolkit without `GraphicShr` variable. A new constructor, which obtains a path to visualization plug-in library (`TKOpenGl.dll` or a redefined one), has been added to `Graphic3d_WNTGraphicDevice`.

22734	*Summary:* Memory allocation error in OpenGl
	
	The following modifications have been introduced to improve memory allocation:
	
	  ▪ `malloc()/free()` have been replaced with `new()/delete()` in `OpenGl_Memory`, `OpenGl_Polygon` and `OpenGl_TextureBox` classes.
	  ▪ Structures and pointers, which served as a vector of elements, have been replaced with `Ncollection_Vector`.
	  ▪ Obsolete class `OpenGl_Memory` has been removed from OCCT.

22751	*Summary:* Issues around `Prs3d_TextAspect::Print()`
	
	  ▪ Unused `Print()` methods enabling to dump the contents of a drawer instance in the console have been removed from `Prs3d` package.
	  ▪ The obsolete code related to unused printing "roles" (`Print`, `PickTraverse` and `Inquire`) has been removed from `OpenGl` and higher-level packages (`Graphic3d`, `Visual3d` and V3d).

22762	*Summary:* Bug in `Graphic3d_Vector::IsParallel` Method `Graphic3d_Vector::IsParallel` has been fixed.
22779	*Summary:* Pixel format should be chosen to support stencil buffer Non-zero stencil depth has been included in the set of search criteria in method `OpenGl_txgl::find_pixel_format` to support stencil buffer.

22782	*Summary:* Uninitialized global variable used in conditionals in OpenGl package
	
	Direct rendering to WinAPI bitmap has ceased to be supported. The following changes have been implemented to remove obsolete workarounds for off-screen rendering from OCCT classes:
	
	  ▪ Unused structure `CALL_DEF_BITMAP` and field `CALL_DEF_VIEW.DefBitmap` have been removed from `InterfaceGraphic` package.
	  ▪ Unused values `OCC_REDRAW_WINDOW`, `OCC_REDRAW_WINDOWAREA`, `OCC_REDRAW_BITMAP` and `OPENGL_NS_ISBITMAP` have been removed from `Aspect_GraphicCallbackProc.hxx`.

Open CASCADE Technology & Products

22787 22906	*Summary:* Gradient background is clipped by planes All clipping planes have been disabled in function `OpenGl_view::TelClearBackground` to avoid problems with display of gradient background.
22795	*Summary:* Possibility to display presentable objects in overlay, grouped by display priority A new option for display of particular objects in overlay groups has been introduced in the viewer. This option is based on a newly designed Z-layer algorithm, which allows locating each object presentation in a so called Z-layer. Each Z-layer has a unique ID generated by method `V3d_Viewer::AddZLayer`. The layers on display are ordered from the lowest (bottom-level layer) to the highest (foreground layer), the sequence of ordered IDs can be returned by `V3d_Viewer::GetAllZLayers` method. The effect of overlay is achieved by clearing depth buffer bits while drawing the next layer. There always exists a default bottom-level layer with ID = 0 that corresponds to the usual display of objects. This layer cannot be removed. When a custom Z-layer is removed, all its objects are put to the default layer. Z-layers are different from the overlayer and the underlayer as the latter are designed to display 2d graphics, while Z-layers are intended to display 3d graphics. The index of a Z-layer can be set for a displayed object by method `AIS_InteractiveContext::SetZLayer`. This method sets Z-layer ID for all computed and displayed object presentations as well as all new presentations that will be added further. The Z-layer IDs for new presentations are maintained by `PrsMgr_PresentationManager`. When all object presentations under the same presentation manager are removed, the default ID value will be used to add a new presentation. New virtual method `PrsMgr_PresentableObject::SetZLayer` accepts the layer ID and `PrsMgr_PresentationManager` as input arguments. This method is intended to control and update all object presentations created both by the presentation manager and by the object itself. This method can be overridden in all custom descendant classes to update the layer IDs of custom presentations. `SelectMgr_SelectableObject` overrides this method to the updated layer IDs for selection presentation. The same method is implemented for `SelectMgr_EntityOwner`.
22796	*Summary:* Possibility to display multi-line text in 3D `OpenGl_Display` text rendering algorithms have been improved to correctly display multi-line text in OCCT 3D view. '\t' is processed specially, i.e. replaced by a fixed number of spaces in addition to '\n' and '\r' characters. Other control characters (like '\b' or '\a') are simply ignored by text rendering code. Standard GL2PS alignment is not used because it does not work correctly for all formats; therefore alignment is calculated manually now.
22844	*Summary:* Misprint in `Select2d_SensitiveSegment` A misprint in `Standard_Boolean Select2D_SensitiveSegment::Matches()` has been corrected. Now this method returns `Standard_True` only when both ends of the segment are inside the selection rectangle.

22879	*Summary:* Problem in `OpenGl_togl_begin_layer_mode.cxx` Method `OpenGl_GraphicDriver::SetTextAttributes` has been redefined to set not only the font name, but also the type of display text and background color for subtitle or decal text display in overlayer or underlayer text. Four different types of display text are available: ▪ `Aspect_TODT_NORMAL` displays text as usual, ▪ `Aspect_TODT_SUBTITLE` displays text on colored background, ▪ `Aspect_TODT_DEKALE` displays text with colored outlines, ▪ `Aspect_TODT_BLEND` displays text with blended colors. The font color itself can be set with method `OpenGl_GraphicDriver::SetColor`. Additionally some modifications have been implemented in V3d package: ▪ Text attributes have been changed from `Aspect_TODT_SUBTITLE` to `Aspect_TODT_NORMAL` in `V3d_ColorScale`. ▪ Text attributes at the beginning of layers redraw have been changed from `Aspect_TODT_SUBTITLE` to `Aspect_TODT_NORMAL` in method `V3d_LayerMgr::Begin()`.
22900	*Summary:* Problem with shape display in shading mode The following modifications have been introduced to avoid problems with shape display in shading mode: ▪ Method `Graphic3d_Group::SetPrimitivesAspect()` has been replaced with method `Graphic3d_Group::SetGroupPrimitivesAspect()` permitting to change aspects without loss of presentation data. ▪ Check if the necessary aspect is already set for the current presentation group has been added in several methods of `AIS_Shape` class to avoid possible loss of presentation data.
22940	*Summary:* Selection performance: TopAbs_SHELL Performance bottleneck in `StdSelect_BrepSelectionTool` class in case of `TopAbs_SHELL` shape type has been corrected. Now this class behaves identically for `TopAbs_SHELL` and `TopAbs_SOLID` shape types, except for different selection priorities used.
22945	*Summary:* `AIS_InteractiveObject`: Initialization bug `AIS_InteractiveObject` class has been modified to properly initialize `myHasTransformation`.
22960	*Summary:* Memory leak in `Xw_get_env.cxx` Obsolete function `Xw_put_env` has been removed from OCCT.
22971	*Summary:* TKOpenGl clean up obsolete functionality The following modifications have been introduced in the frame of this improvement: ▪ Duplicate primitive types `PolygonHoles`, `QuadrangleMesh`, `TriangleMesh`, `TriangleSet` and `Bezier` have been removed from `OpenGl_GraphicDriver`, `Graphic3d_GraphicDriver` and `Graphic3d_Group` classes ▪ `Polyline` is temporary redirected to `PrimitiveArray` in `Graphic3d_Group`. ▪ Unused class `StdPrs_ShadedPolSurface` has been removed. ▪ Use of primitive arrays has been enabled in `AIS_TexturedShape`. `Prs3d_ShadedShape` texture coordinates functionality has been added to `StdPrs_ShadedShape` to eliminate code duplication.

22973 23032		*Summary:* Crash of Draw command `voxeloctboolds` on MMGT_OPT=0, MMGT_REENTRANT=1 Method `Voxel_OctBoolDS::OptimizeMemory()` has been improved to avoid possible crash.
23000		*Summary:* Improved handling of gradient and textured background in 3d viewer Management of gradient and textured background in the 3d view has been changed. `V3d_View::SetBgImageStyle()` function enables to display a background image (in the centered mode) above the gradient background or to clear it from the view. The following options are possible: ▪ `Aspect_FM_NONE` – the texture is not rendered, however, it is not removed and remains available in the viewer, so it is possible to redisplay it using `SetBgImageStyle()` function. ▪ `Aspect_FM_CENTERED` – the texture is displayed in the center of the view. ▪ `Aspect_FM_TILED` – the texture is tiled in the view. ▪ `Aspect_FM_STRETCH` – the texture is stretched to the view size. The display of gradient background has been extended. The following options are possible for `V3d_View::SetBgGradientStyle()` function: ▪ `Aspect_GFM_NONE` – the gradient background is not drawn. ▪ `Aspect_GFM_HOR` – the gradient is rendered from the left (color1) to the right (color2). ▪ `Aspect_GFM_VER` – the gradient is rendered from top (color1) to bottom (color2). ▪ `Aspect_GFM_DIAG1` – the gradient is rendered from the top-left corner (color1) to the bottom-right corner (color2).`Aspect_GFM_DIAG2` – the gradient is rendered from the top-right corner (color1) to the bottom-left corner (color2). ▪ `Aspect_GFM_CORNER1` – the gradient is rendered from the top-left corner (color1) to other corners (color2). ▪ `Aspect_GFM_CORNER2` – the gradient is rendered from the top-right corner (color1) to other corners (color2). ▪ `Aspect_GFM_CORNER3` – the gradient is rendered from the bottom-right corner (color1) to other corners (color2). ▪ `Aspect_GFM_CORNER4` – the gradient is rendered from the bottom-left corner (color1) to other corners (color2). The following fixes have been added: ▪ `V3d_View::myBackground` data member is now properly changed by `V3d_View::SetBgGradientStyle()`. ▪ Gradient background is now properly drawn even if `V3d_View::SetBgGradientStyle(<any_fill_type>)` is called after `V3d_View::SetBgGradientStyle(Aspect_GFM_NONE)`. The following Draw commands have been added to test background operations: ▪ `vsetcolorbg` sets solid background: `V3d_View::SetBackgroundColor()`; ▪ `vsetgrbgmode` switches gradient background fill type: `V3d_View::SetBgGradientStyle()`; ▪ `vsetbgmode` switches texture background fill type: `V3d_View::SetBgImageStyle()`.

23005	*Summary:* Unjustified memory occupation during undo/redo operation NIS package has been revised: ▪ The vector type of interactive context objects collection has been replaced by sparse array to free allocated memory when an object is removed in `NIS_InteractiveContext`. ▪ Drawers without associated interactive objects have been removed from interactive context in `NIS_Drawer`. ▪ `NIS_ObjectsIterator` has been improved to remove deleted interactive objects from all structures of interactive context.
23012	*Summary:* Detection gives incorrect results The tolerance management in selection algorithm has been improved. ▪ If `StdSelect_ViewerSelector3d::SetSensitivityMode()` is set to `StdSelect_SM_WINDOW` mode, the pixel tolerance is specified through `StdSelect_ViewerSelector3d::Set(const Standard_Integer)`, so it is fixed and independent from the current zoom level. When the scene is zoomed out, the 3D tolerance used by the detection algorithm is calculated from this pixel tolerance. In this case some thin shapes are processed incorrectly as for them the 3D tolerance can be greater than a half of the shape thickness. ▪ If the new mode `StdSelect_SM_VIEW` is chosen, the sensitivity used by the viewer selector is not recalculated, `myupdatetol` flag remains `Standard_False` and the sensitivity is set directly using `SelectMgr_ViewerSelector::SetSensitivity()`. Since the sensitivity is measured in the model coordinates, it is always possible to determine the value of this parameter relevant for the correct selection basing on the bounding box coordinates of the displayed shapes. For contiguous shapes, this value should be slightly less than the half of the least value of all dimensions (dx, dy, dz) of the shapes. Other useful methods are: ▪ `StdSelect_ViewerSelector3d::PixelTolerance()` which returns the current pixel tolerance value and ▪ `SelectMgr_ViewerSelector::Sensitivity()`, which returns the current sensitivity value. The methods have been also implemented in `AIS_InteractiveContext`, which redirects the request to the opened local context and `AIS_LocalContext` classes, which redirects the request to the selectors. New command `vselprecision [precision_mode [tolerance_value]]` has been added to change sensitivity mode and tolerance value in Draw. ▪ `precision_mode` can take value 0, which corresponds to `StdSelect_SM_WINDOW` mode, or 1, which corresponds to `StdSelect_SM_VIEW`; ▪ `tolerance_value` is the integer value of pixel tolerance (if `precision_mode=0` or double value of sensitivity (if `precision_mode=1`) Called without arguments, the command prints the current mode and tolerance.
23022	*Summary:* Possibility to access `OpenGl` extensions and core API (1.2+) in one place `OpenGl_Context` class has been revised and improved to provide OpenGL extensions and core API up to version 2.0 in one place. Chaotic inclusion of '`<GL/gl.h>`' header has been eliminated as well.

23044		*Summary:* 3D views are not invalidated by some modifications of `OpenGl_Structure` `Update()` methods in `V3d_View` and `V3d_Viewer` classes have become obsolete. The corresponding `Redraw()` methods should be used instead.
23064		*Summary:* MSVC compiler warnings when `FreeImage` is not used Compiler warning in `OpenGl_Workspace.cxx` on conversion of BOOL to bool generated when HAVE_FREEIMAGE is not defined has been avoided.

23067		*Summary:* `OpenGl` package API needed by `UserDraw` feature is not exported
		
		`OpenGl` package API has been made available to applications that implement custom scene elements with help of `UserDraw` feature:
		  ▪ Inline method providing access to `OpenGl_Context` instance has been added in `OpenGl_Window` class.
		  ▪ `OpenGl_Workspace` class exports `AspectXXX()` methods returning current visual parameters for different kinds of primitive and inline methods for text rendering.
		`OpenGl_ResourceCleaner`, `OpenGl_ResourceTexture` and `OpenGl_ResourceVBO` classes can be used to clean-up `OpenGL` resources safely.
		`OpenGl_AspectFace`, `OpenGl_AspectLine`, `OpenGl_AspectMarker` add `OpenGl_AspectText` classes read access to all attributes with help of inline methods. Other state parameters should be obtained with help of `OpenGL` API calls.
		
		DRAW command `vuserdraw` has been added as a source code example illustrating how `UserDraw` functionality could be used in application.

23069		*Summary:* OpenGL state extraction for GL_RENDER_MODE significantly impacts performance GL_RENDER_MODE checking in `TKOpenGl` routines for primitive arrays and markers rendering is now avoided to improve rendering performance.
23081		*Summary:* Retrieve GPU memory information from graphic driver New method `Graphic3d_GraphicDriver::MemoryInfo()` has been created to provide system-dependent GPU memory usage information (only for AMD/Ati and NVIDIA drivers). Notice that in most cases the values are obtained for the entire system and they are not process-restricted so applications running in parallel, especially 3D ones, affect the results. New class `OSD_MemInfo` provides information about process memory consumption (virtual memory, working set). These memory tools are no replacement for system monitors and memory analysis tools. They are intended to perform detailed memory reports for concrete code intervals (defined at compile time) where traditional tools are not applicable or less efficient. Draw commands `vmemgpu` and `meminfo` provide access to these tools from `tcl` environment allowing to easily retrieve memory information at any time and to perform automatic tests when applicable.
23101		*Summary:* `TKOpenGl` possible `glext` header conflicts Macro GL_GLEXT_LEGACY has been introduced to avoid the conflict between old and new `<GL/glext.h>` definitions.

23102	*Summary:* Change the algorithm of 3D viewer background rendering using tiled texture 3D viewer background textures are now tiled starting from the top-left corner of the view (instead of the bottom-left corner as earlier).
23115	*Summary:* Polygon offset doesn't applied in Viewer3D sample Method `AIS_InteractiveObject::SetPolygonOffsets()` has been improved to update all `Graphic3d_Group` instances of an object that has `Graphic3d_AsepctFillArea3d` aspect set. Without this, polygon offset parameters are changed for `Graphic3d_Structure` only, which has no effect. DRAW command `vpolygonoffset` has been added to set or print polygon offset parameters for a displayed object or to print the default values.

Application Framework

22685	*Summary:* Missing parentheses around `trigraph` operator The parentheses have been corrected in several classes of `Aspect` and `TDataStd` packages.
22693	*Summary:* Eliminate Compiler warnings The order of operations has been specified explicitly in `TNaming_Localizer.cxx`.
22742	*Summary:* Not all constructors defined in `TNaming_Identifier.cdl` are implemented. The second constructor of `TNaming_Identifier` class has been correctly implemented in `TNaming_Identifier.cxx`.
22788 22874	*Summary:* Exception during closing of document due to corrupted memory Classes TNaming_NamedShape and TNaming_UsedShapes have been improved to provide correct destruction of the document containing a shape during operation "Close". A work-around previously used to address this problem has been removed from TObj_Model class.
22943	*Summary:* Problem with `TDataXtd_PatternStd` A wrong index has been fixed in class `TDataXtd_PatternStd`.
22959	*Summary:* Accessing uninitialized variable (pointer): `aCurElement` Class `XmlMDataStd_NamedDataDriver` has been fixed to avoid warnings.
22995	*Summary:* It is not possible to check presence of a document in OCAF session via `IsInSession()` on Windows Method `TDocStd_Application::IsInSession()` has been modified to allow checking presence of a document in OCAF session.

23002	*Summary:* Empty delete operator in TDF_Label Node Destruction of TDF_Label Node has been corrected to use consistently defined new/delete operators
23085	*Summary:* Call of tcl DFBrowser leads to error message Command DFBrowse has been improved to provide correct loading of DFBrowser. Now script dftree.tcl is loaded inside the command.

Data Exchange

15697	*Summary:* Contribution to OCC Data Exchange Improvement
	
	The following improvements have been added:
	  ▪ Protection on null geometric set has been implemented in STEP reader (StepToTopoDS_TranslateShell class).
	  ▪ Protection against invalid color values while reading IGES has been implemented in IGESCAFControl_Reader.
	  ▪ igesread.c. has been protected against possible incorrect terminal section.

22092	*Summary:* Crash of application loading a VRML file with all degenerated triangles Protection against crash at loading of a VRML file containing all degenerated triangles has been added to OCCT in VrmlData package.
22165	*Summary:* IGES translator does not produce shape Report messages shown in case of a translation error have become more informative.
22459	*Summary:* ImportExport sample crash The method for translation of a composite curve from STEP file to TopoDS (StepToTopoDS package) has been improved. If one or more segments of a composite curve are lines, a wire built from the curve is invalid (FixReorder() and FixConnected() failed) and is transferred to the result shape as a compound.
22543	*Summary:* Attempt to fix some errors reported by valgrind at Draw execution Operator delete has been added for a_string in Draw_Interpretor.cxx to fix possible memory leak.
22728	*Summary:* STEP export API does not provide possibility to control output stream STEP file export procedure has been extended with a specific check if the disk where the file is exported to is write protected or has no space for the file. This check implemented in class IFSelect_WorkSession allows distinguishing between such failure causes and object translation problems.
22731	*Summary:* Error on attempt to write in IGES 5.3 the attached brep model BRepToIGESBRep_Entity and IGESControl_Writer have been fixed to avoid translation error.

22776	*Summary:* XCAFPrs_AISObject does not support transparency The problem caused by redefinition of transparency setting in method XCAFPrs_AISObject::Compute has been fixed.
22805	*Summary:* Bug of STEP read /writer The algorithm of translation of an edge loop from STEP file to OCC has been modified in class StepToTopoDS_TranslateEdgeLoop. If the computation of a 2D parameter on edge gives the same result, the range of edge isn't updated.
22807	*Summary:* Loading of STEP entities in model during reading of STEP file requires redundant memory The following changes and modifications have been introduced in OCCT to improve memory management: ▪ Fields "theshareds" and "thesharnews" have been removed from class Interface_Graph. ▪ The field type "thesharings" in this class has been changed from Interface_IntList to TColStd_Harray1ListOfInteger. ▪ Fields "thestats" and "theflags" in class Interface_Graph are filled only in the case specified by mode from class IFSelect_WorkSession (can be necessary for selections and not used for reading STEP in application) ▪ Handle(Standard_Transient) has been replaced by Handle(Standard_Type) in fields "thentnum" and "thentmod" from Interface_Gtool, which allows reducing map size from 2400000 to 62. ▪ Additional mode for filling fields "thestats" and "thefalgs" from Interface_Graph has been implemented in IFSelect_WorkSession. ▪ The type of field "theidnums" has become TcolStd_Harray1OfInteger in class StepData_StepModel. ▪ Classes XSControl_WorkSession and Transfer_TransientProcess have been modified to make method XSControl_WorkSession::NewModel() completely clear of memory ▪ Optional flag has been added in command "stepread" from XSDRAWSTEP. This flag defines the mode to be used (full or reduced model type). By default this mode is set to false.
22822	*Summary:* Skipping of the first and the last edges of a wire for opened wires The method ShapeAnalysis_Wire::CheckSelfIntersection() has been improved to correctly process the first and the last edges of an opened wire.
22866	*Summary:* Now it is not possible to mesh a shape in parallel mode during export to STL It has become possible to mesh shape in parallel mode during export to STL.
22915	*Summary:* Crash in IGESControl_Reader::TransferRoots() Method IGESGeom_CircularArc::IsClosed() has been modified to take into account Precision::Pconfusion() during the verification on closure.
22934	*Summary:* Wrong delete operator in IGESSelect_SelectFromDrawing.cxx / IGESSelect_SelectFromSingleView.cxx Obsolete code has been removed from classes IGESSelect_SelectFromDrawing and IGESSelect_SelectFromSingleView.

22941	*Summary:* Memory leak in `StepData_StepReaderData.cxx` Potential memory leak in `StepData_StepReaderData` class has been fixed by replacing C pointers by array classes from `TColStd` package.	
22947 22948	*Summary:* `XCAFPrs_AISObject.cxx: array index out of bounds …` `XCAFPrs_AISObject` class has been corrected by re-using the implementation of method `AIS_Shape::DisplayBox()` instead of a copy.	
22961	*Summary:* Dangerous usage of `'buf'` (`strncpy` doesn't always 0-terminate it) The following modifications have been implemented in classes `VrmlData_Group` and `VrmlData_Scene`: ▪ Zero-termination has been added for "buf" variable; ▪ `char` arrays have been replaced by `Tcollection_AsciiString`.	
22962	*Summary:* Invalid realization of reading and writing material in STEP Classes `STEPCAFControl_Reader` and `STEPCAFControl_Writer` have been corrected to properly read and write material data in STEP.	
22982	*Summary:* Generic color is overridden in STEPCAFControl_Writer::WriteColors Logic of writing colors in STEP file has been fixed to deal correctly with overridden colors.	
22993	*Summary:* Crash during reading STEP file Class `STEPControl_ActorRead` has been corrected to avoid exception when the header of the read STEP file is invalid.	
23009	*Summary:* Request of a new feature for units management during step import New method `STEPControl_Reader::FileUnits()` returns sequences of unit names for LENGTH_UNIT, PLANE_ANGLE_UNIT and SOLID_ANGLE_UNIT. To get these units from a file, it is necessary to read the file in memory with method `ReadFile()` and to apply method `FileUnits()` before transfer to BRep representation. The corresponding Draw command `stepfileunits file_name` prints in the console the names of all units found in the file.	
23016	*Summary:* Elimination of dependency of Tcl OCAF Browser from Tix product Tcl OCAF Browser has been updated to work with `Tcl/Tk 8.5` without dependency on Tix. Tcl code has been simplified and C++ code has been extended to support more attributes.	
23023	*Summary:* VRML reader fails on attempt to read an attached WRL file Method `VrmlData_TextureCoordinate::Read` has been improved to allow various positions of the closing square bracket in the translated file.	

Shape Healing

22746	*Summary:* Progress indicator in ShapeHealing Progress indicator Message_ProgressIndicator has been implemented in methods of ShapeFix package (first of all, ShapeFix_Shape::Perform()) to indicate operation progress.
22757	*Summary:* Exception in ShapeHealing Method ShapeAnalysis::AdjustByPeriod has been corrected to use "floor" for rounding instead of conversion to Standard_Integer, to allow safe processing of big real numbers.
22848	*Summary:* Optimize projection of points in ShapeAnalysis_Surface Projection of points in ShapeAnalysis_Surface made by class Extrema_GenExtPS has been optimized by avoiding calculation of MAX extrema (recently added flag).
22873	*Summary:* Correction in ShapeConstruct_ProjectCurveOnSurface.cxx Check for zero distance in methods CheckPoints() and CheckPoints2d() has been improved as comparison with exact zero and taking sqrt are avoided.
22924	*Summary:* ShapeAnalysis_FreeBounds::ConnectEdgesToWires() is unstable Class ShapeAnalysis_BoxBndTree has been modified to keep the order, in which the source edges are given in the input list, in the resulting wires.

Draw

20750	*Summary:* DrawAppliInit file from the current directory is used during Draw starting DrawAppliInit file is now loaded from ros by default or not used at all when Draw application is launched without CSF_DrawPluginDefaults variable set.
21690	*Summary:* "cpulimit" command is not implemented on Win32 platform Command cpulimit has been implemented on Windows platform. Command "limitelapsed" has been removed from package SWDRAW as obsolete.
22316	*Summary:* Mistakes in TestTopOpeDraw_OtherCommands.cxx Function <strcmp ()> is now used instead of "==" to compare char* strings in class TestTopOpeDraw_OtherCommands.
22378	*Summary:* Fix of buffer vulnerability in Draw Buffer vulnerability has been fixed in class Draw_Window.

22483	*Summary:* ZBufferTrihedron loses user-defined color Draw command vzbufftrihedron has been extended to the following set of arguments: Position, textR, textG, textB, Scale and Type_Of_Visualization. If the command is used without any parameters, a default trihedron will be displayed at the bottom left corner of the view.
22682	*Summary:* Draw options -v and -f do not influence on AISInitViewer command ViewerTest_Tool has been modified to avoid problems with incorrect viewer size.
22726	*Summary:* Exception on restoring shape Safe-check in DBRep_IsoBuilder class has been improved to correctly define the interval in all cases.
22767	*Summary:* Extension of Draw command fixshape Classes ShapeFix_Face, ShapeFix_Shell and ShapeFix_Wire have been modified to remove duplicate messages. Most messages now attached to the shape, which has been fixed. Necessary breaks have been added to the switch case in SWDRAW_ShapeFix class Statement fixshape (line 389) now provides information about the number of fixes if called with option '+?' and stores all modified shapes in compound if called with '*?'
22770	*Summary:* Improve command sewing in Draw Draw command sewing has been improved in class BRepTest_SurfaceCommands to provide all options implemented in the sewing algorithm
22799	*Summary:* BrepMesh testing improvement ▪ mpparallel and mpincmesh Draw commands have been removed; ▪ IsParallel flag has been added as a third argument into incmesh command. If IsParallel is set, meshing is done in a parallel mode.
22847	*Summary:* Provide interface in DrawTrSurf New Draw commands have been introduced for configuring visualization properties in axonometric 2D and 3D viewers: ▪ setpointcolor - sets global color for all consequently created points; ▪ changepointcolor - sets color for a given point; ▪ setpointmarker - sets global marker for all consequently created points; ▪ changepointmarker - sets marker for a given point;
22975	*Summary:* Help for Draw command ImportShape is incorrect. Help message for Draw command ImportShape has been corrected.
22976	*Summary:* Draw command CopyDF fails to copy content of a label Draw command CopyDF has been modified in DDF_DataCommands.cxx to successfully copy real attribute from label 0:1 to label 0:2.

23030	*Summary:* Collect all DRAW commands created for testing bugs into one package Draw commands for testing of specific project bugs have been grouped into one QABugs package.
23065	*Summary:* DRAW command to estimate visualization performance The following new commands have been implemented to test visualization performance: ▪ `vfps` performs several (100 by default) redraw calls of the active 3D view and returns two values: "Average FPS" and "Average CPU (user) time". Average FPS represents overall performance (can be limited by the upper-bound due to some factors, such as VSync). CPU time can be used to determine if the performance is CPU-limited or GPU-limited. ▪ `vvbo` controls VBO usage flag (turned ON by default). This command is used to check only the backward compatibility rendering flow. There are no reasons to disable VBO on modern systems due to significant performance impact. VBO controls and time measurements have been removed from command `vdrawsphere`. This command generates an object with variable complexity of geometry to estimate pure rendering performance.
23110	*Summary:* `ViewerTest` dependency on `OpenGL` libraries Dependency of `ViewerTest` package on `OpenGL` libraries has been added to `EXTERNLIB`.

Documentation

22865	*Summary:* Lost reference information in OCCT6.5.2 documentation generated by a new version of Doxygen The reference documentation has been built with doxygen 1.7.4 to provide information about the module, toolkit and package to which a class belongs on class reference page
22870	*Summary:* Visualization User's Guide of OCC 6.5.2 has damaged images. Missing images have been restored in the Visualization User's Guide.
23045	*Summary:* Add Readme file to OCCT sources A Readme file providing basic information on how to start working with OCCT has been placed in the root folder of the repository.

Open CASCADE Technology & Products

Development Environment

22913 22922 23041	*Summary:* Clean up warnings on uninitialized / unused variables OCCT has been overhauled to remove or comment out uninitialized or unused variables, which produced compilation warnings.
23024	*Summary:* Update headers of OCCT files Headers of OCCT files have been updated to correspond to the current status of OCCT.

WOK

22613	*Summary:* Retrieve environment variable in EDL EDL parser now determines `${EnvVarName}` syntax to automatically retrieve environment variable. Only the simplest syntax is supported (no `"text${EnvVar}/${anotherVar}"`, `"$EnvVarName"`, or such). Usage example: `@set ENV_MYVAR = "${EnvVarName}"`.
23107	*Summary*: Update of OCCT service files for Makefile files generation Service files of OCCT packages `OS` and `QADraw` and toolkit `TKQADraw` have been updated for correct generation of `Makefile` files by WOK.

Release

22752	*Summary:* Fix compilation on Unix with `FreeImage` and GL2PS support Includes of `config.h` have been added to some classes from `OpenGl` package and `Image_PixMap.cxx` to build `TKOpenGl` with enabled 3rd-party components using `Makefiles`.
23013	*Summary:* Fails to build with gcc 4.6.3 Unnecessary cast in `BRepClass3d_SClassifier::PerformInfinitePoint` has been removed to provide correct building of OCCT with gcc 4.6.3.

MFC samples

22800	*Summary:* `OSD::SetSignal (Standard_True)` is not called in the standard samples. Call `OSD::SetSignal (Standard_True)` has been added in the constructor of `OCC_BaseApp` to avoid failures with reading of STEP and IGES files.
23097	*Summary*: MFC Samples do not compile after redesigning the `TKOpenGl` driver Obsolete `TriangleSet` rendering primitive has been replaced by the corresponding sub-class of `Graphic3d_ArrayOfPrimitives` to visualize triangles in `User_Cylinder`.

Open CASCADE Technology & Products

Products

Advanced Samples

| 22756 | *Summary:* Button missing on Windows 7

The interface of XDE Sample has been improved to correctly display button "Options" in "STL Export" dialog both on Windows XP and Windows 7. |
| --- | --- |

DXF Export

| 22996 | *Summary:* Exception at writing to DXF

Incorrect type cast that caused an exception in binaries built with MS VC 10 has been fixed. The code for writing time and date has been modified to use function `Interface_MSG::TDate`. |
| --- | --- |

ACIS Import

| 22707 | *Summary:* Invalid shape translating a SAT file

ShapeFix_Face tool has been extended with method `FixPeriodicDegenerated` giving the possibility to reconstruct the face boundaries when a single periodic wire is specified on a conical surface. The missing boundary is generated in the apex of the cone and the wire is properly oriented. |
| --- | --- |
| 22748 | *Summary:* Invalid shape translating a SAT file

A translation problem caused by incorrect ratio coefficient (`ratio > 1`) of some `'ellipse-curve'` entities has been fixed by modification of `AcisData_MakeCurve` - all such cases are processed as circles (`ratio = 1`). |

Express Mesh

| 22128 | *Summary:* Incorrect mesh

The following improvements have been introduced in Express Mesh:
- `QMShape_Tessellator::DiscretiseEdge` method has been adjusted to check the same parameterization for each passed edge regardless of `SameParameter` flag from `BRep_TEdge`. If the number of points in 2D and 3D polygons for a certain edge is not the same that edge goes to the collection of bad shapes, and the discretisation is not performed for it.
- Method `QMShape_Tessellator::DiscretiseCurve` has been applied not only for discretisation of 3D curve belonging to an edge, but also if a problem with parameterization is detected.
- Method `QMShape_DiscrCurve::CheckDeflection2d` has been modified to recognize the edges are not `SameParameter` and process them by projection. The projection is now performed by means of `ShapeAnalysis_Surface` rather than by `GeomAPI`. |
| --- | --- |

22129	*Summary:* Empty triangulation on face Method `QMTools_WireCorrector::KnitEdges2d` has been modified to remove single-segment loops from discrete model. Method `QMTools_WireCorrector::Perform` has been modified to reconstruct discretization wires correctly after such removal.
22775	*Summary:* Cannot build a mesh for the attached face with thin pin on one of corners Classes `QMData_Curve` and `QMTools_WireCorrector` have been modified to enable the meshing algorithm to build a correct triangulation of faces with self-intersecting boundaries, if the area of intersection is negligibly small.
22889	*Summary:* Replace `NCollection_Sequence` class by another container to avoid data races in parallel version of Express Mesh `NCollection_Sequence` container for PCurve representation has been replaced by `NCollection_List` in `QMData_Edge` class. Correspondingly, the methods working with myPCurves field in `QMData _Edge` class have been changed.
22949	*Summary:* Incorrect polygon in 2D space New method `QMTools_WireCorrector::RemoveParametricallyDegenerated` has been implemented to remove discrete segments which have negligible 3D and 2D representations. The merging algorithm in class `QMTools_WireCorrector::KnitEdges2d` has been modified to avoid self-intersection problems. Self-intersection check `QMTools_WireCorrector::IsIntersectingSeg` and a special cutting procedure `QMTools_WireCorrector::RemoveIntersectionLoop`, which removes intersection loops if they are small enough, have been implemented.

Open CASCADE Technology & Products

Porting to version 6.5.3

Porting of user applications from the previous 6.5.2 OCCT version to version 6.5.3 requires the following issues to be taken into account:

- As a result of code clean-up and redesign of `TKOpenGl` driver, some obsolete functions and rendering primitives (`TriangleMesh`, `TriangleSet`, `Bezier`, `Polyline`, `Polygon`, `PolygonHoles`, `QuadrangleMesh` and `QuadrangleSet`) have been removed. Instead, the application developers should use primitive arrays that provide the same functionality but are hardware-accelerated. The details can be found in OCCT Visualization User's Guide, "Primitive Arrays" chapter.
- Applications should not call `AIS_InteractiveObject::SetPolygonOffsets()` method for an instance of `AIS_TexturedShape` class after it has been added to `AIS_InteractiveContext`. More generally, modification of `Graphic3d_AspectFillArea3d` parameters for the computed groups of any `AIS_InteractiveObject` subclass that uses texture mapping should be avoided, because this results in broken texture mapping (see issue 23118). It is still possible to apply non-default polygon offsets to `AIS_TexturedShape` by calling `SetPolygonOffsets()` before displaying the shape.
- The applications that might have used internal functions provided by `TKOpenGl` or removed primitives will need to be updated.
- In connection with the implementation of Z-layers it might be necessary to revise the application code or revise the custom direct descendant classes of `Graphic3d_GraphicDriver` and `Graphic3d_StructureManager` to use the Z-layer feature.
- Global variables `Standard_PI` and `PI` have been eliminated (use macro `M_PI` instead).
- Method `HashCode()` has been removed from class `Standard_Transient`. It is advisable to use global function `::HashCode()` for `Handle` objects instead.
- Declaration of operators `new`/`delete` for classes has become consistent and is encapsulated in macros.
- Memory management has been changed to use standard heap (`MMGT_OPT=0`) and reentrant mode (`MMGT_REENTRANT=1`) by default.
- Map classes in `NCollection` package now accept one more argument defining a hash tool.

Open CASCADE Technology & Products

Supported Platforms and Pre-requisites

Open CASCADE Technology is supported on Windows Intel and Linux Intel platforms.

The table below lists the product versions used by OCCT and its system requirements.

Linux Operating System	32/64-bit: Debian 4.0, Mandriva 2008*
Windows Operating System	32/64-bit: MS Windows SEVEN SP1 / VISTA SP2 /XP SP3
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
Minimum swap space	500 Mb
Video card	**GeForce** The following versions of GeForce drivers are recommended: *For Linux:* 64-bit Version: 100.14.19 or later 32-bit Version: 100.14.19 or later *For Windows:* Version 266.58 WHQL or later is recommended: http://www.nvidia.com/Download/index.aspx
Graphic library	OpenGL 1.1+
C++	*For Linux:* GNU gcc 4.0. - 4.3.2. *For Windows:* Microsoft Visual Studio .NET 2005 SP1** with all security updates Microsoft Visual Studio .NET 2008 SP1 Microsoft Visual Studio .NET 2010
TCL (for testing tools)	*For Linux:* Tcltk 8.5 http://www.tcl.tk/software/tcltk/8.5.html *For Windows:* ActiveTcl 8.5 http://www.activestate.com/activetcl/downloads
Qt (for demonstration tools)	Qt 4.6.2 http://qt.nokia.com/downloads
Freetype (OCCT Text rendering)	freetype-2.3.7 http://sourceforge.net/projects/freetype/files/
Ftgl (OCCT Text rendering)	ftgl-2.1.2 http://sourceforge.net/projects/ftgl/files/
FreeImage* (Support of common graphic formats)	FreeImage 3.14.1 http://sourceforge.net/projects/freeimage/files/Source%20Distribution/
gl2ps* (Export of OCCT viewer contents to vector graphic file)	gl2ps-1.3.5 http://geuz.org/gl2ps/
TBB* (Tool for parallelized version of BRepMesh component)	tbb30_018oss http://www.threadingbuildingblocks.org/

- * Mandriva 2010 is a permanently tested platform.
- ** The official release of OCCT for Windows contains libraries built with VC++ 2008.
- *** This product is optional.