Open CASCADE Technology and Products ver. 7.0.0 Major Release

Release Notes

Overview

Open CASCADE Technology and Products version 7.0.0 is a major release, which includes more than **500** new features, improvements and bug fixes over maintenance release 6.9.1.

Version 7.0.0 is binary and source incompatible with the previous versions of Open CASCADE Technology and Products, so applications linked against a previous version must be ported and recompiled to run with this Version 7.0.0.

Highlights

Configuration

- Conversion of CDL classes to plain C++
- Use of CMake as main build system, replacing WOK

Foundation Classes

- New implementation of shared pointer (Handle)
- Redesign of OCCT type system

Modeling algorithms

- Refactoring of B-Spline evaluation
- Ability to perform Boolean expressions on an arbitrary number of arguments
- Intersection of surfaces produces more accurate b-spline curves
- More predictable offset of 3D shapes with a large offset value in “Intersection” mode

Visualization

- Activation of selection modes without opening the local context
- OpenGL graphic rendering methods exposed to the client code
- Support of zoom persistent selection
- Configurable font orientation
- Direct3D integration toolkit
- Support of antialiasing using multisampling technique (MSAA)

Application framework

- Interface for reading / saving documents from / to arbitrary C++ stream
- Main OCAF toolkits are made independent on Visualization toolkits

Data exchange

- Support of reading and writing semantic PMI entities for STEP AP242 format
- Optimization of shape triangulation export to STL and VRML

Products

- ACIS SAT Import-Export: new geometrical entities and color attributes supported
- DXF Import-Export: improved translation of UNICODE text
- BestFit: improved calculations with non-null offset on two-sided models
- Mesh Framework Kernel: accelerated interfaces to STL and OBJ formats
- Advanced Samples: new WPF Import/Export sample added to the Advanced C# Wrapper tool

Table of Contents

New features

Template-based Handles

OCCT Handle classes are now implemented as C++ template, `opencascade::handle<>`. It can be used with any class inheriting (directly or indirectly) `Standard_Transient`.

Traditional pre-processor macros are still provided for convenience and used to deal with handles in OCCT code:

- `Handle(Class)` expands to `opencascade::handle<Class>`.
- `DEFINE_STANDARD_HANDLE(Class,Base)` defines name `Handle_Class` as `typedef` to `opencascade::handle<Class>`

The existing interface has been preserved in general, thus handles can be used in the same way as before in most situations. See Upgrade Guide for description of possible incompatibilities.

Arbitrary Boolean Expressions or Edition of General Fuse results

It has become possible to add or remove any part to/from the result of a General Fuse Boolean operation and to remove any internal boundaries between parts.

The result of the new operation is a compound containing selected parts of the basic type (`VERTEX`, `EDGE`, `FACE` or `SOLID`). The default result is an empty compound. It is possible to add any split part to the result by using methods `AddToRessult`() and `AddAllToResult`().

It is also possible to remove any part from the result by using methods `RemoveFromResult`() and `RemoveAllFromResult`(). The method `RemoveAllFromResult`() is also suitable for clearing the result.

It is possible to use method `RemoveInternalBoundaries`() to unite all parts with the same material. The material should not be equal to 0, as this is the default material value. The boundaries between parts with this value will not be removed.

It is not possible to unite two parts that have different materials. To remove boundaries during combining the result, define the material for parts (not equal to 0) and set the flag `bUpdate` to `TRUE`. However, for the `FACE` or `EDGE` arguments it is recommended to remove the boundaries in the end when the result is completely built. It helps to avoid self-intersections in the result.

It is possible to create typed Containers from the parts added to result using method `MakeContainers`(). The type of the containers will depend on the type of the arguments: `WIRES` for `EDGES`, `SHELLS` for `FACES` and `COMPSOLIDS` for `SOLIDS`. The result will be a compound containing containers.

Addition of parts to this result will not update containers. The result compound will contain the containers and newly added parts (of basic type). Removal of the parts from this result may affect some containers if the parts that should be removed are in the container. In this case this container will be rebuilt without them.

The algorithm supports history information available through methods `IsDeleted`() and `Modified`(). In DRAW Test Harness it is available through the same commands as for Boolean Operations (`bmodified` and `bisdeleted`).

Example of the API:

```
BOPAlgo_CellsBuilder aCBuilder;
BOPCol_ListOfShape aLS = ...; // arguments
/* parallel or single mode (the default value is FALSE)*/
Standard_Boolean bRunParallel = Standard_False;
/* fuzzy option (default value is 0)*/
Standard_Real aTol = 0.0;
//
aCBuilder.SetArguments(aLS);
aCBuilder.SetRunParallel(bRunParallel);
aCBuilder.SetFuzzyValue(aTol);
//
aCBuilder.Perform();
if (aCBuilder.ErrorStatus()) { // check error status
   return;
    }
/* empty compound, as nothing has been added yet */
    const TopoDS_Shape& aRes = aCBuilder.Shape();
/* all split parts */
    const TopoDS_Shape& aRes = aCBuilder.GetAllParts();
  //
BOPCol_ListOfShape aLSToTake = ...; // parts of these arguments will be taken into result
BOPCol_ListOfShape aLSToAvoid = ...; // parts of these arguments will not be taken into result
//
/* defines the material common for the cells, i.e. the boundaries between cells with the same material that
will be removed. By default it is set to 0. Thus, to remove a boundary the value of this variable should not be
equal to 0 */
Standard_Integer iMaterial = ...;
/* defines whether to update the result right now or not */
Standard_Boolean bUpdate = ...;
  // adding to result
aCBuilder.AddToResult(aLSToTake, aLSToAvoid, iMaterial, bUpdate);
aR = aCBuilder.Shape(); // the result
  // removing of the boundaries
aCBuilder.RemoveInternalBoundaries();

  // removing from result
aCBuilder.AddAllToResult();
aCBuilder.RemoveFromResult(aLSToTake, aLSToAvoid);
aR = aCBuilder.Shape(); // the result
```

Example of new operation in DRAW

```
psphere s1 15
psphere s2 15
psphere s3 15
ttranslate s1 0 0 10
ttranslate s2 20 0 10
ttranslate s3 10 0 0

bclearobjects; bcleartools
baddobjects s1 s2 s3
bfillds
  # rx will contain all split parts
    bcbuild rx
  # add to result the part that is common for all three spheres
    bcadd res s1 1 s2 1 s3 1 -m 1
  # add to result the part that is common only for first and third spheres
    bcadd res s1 1 s2 0 s3 1 -m 1
  # remove internal boundaries
    bcremoveint res
```

Exposing the interface of OpenGl_View

The interface of OpenGl_View (OpenGL graphics rendering methods) has been exposed to the client code and all high-level API methods of application views have been collected in V3d_View class.

OpenGl_View interface:

The class `OpenGl_View` now inherits from new class `Graphic3d_CView`.
`Graphic3d_CView` is an interface class that declares abstract methods for managing displayed structures, display properties and a base layer code that implements computation and management of HLR (or more broadly speaking view-depended) structures.

In the new implementation it takes place of the eliminated `Visual3d_View`. As earlier the instance of `Graphic3d_CView` is still completely managed by `V3d_View` classes. It can be accessed through `V3d_View` interface but normally this should not be required as all its methods are completely wrapped.

In more details, a concrete specialization of `Graphic3d_CView` is created and returned by graphical driver on request. Right after creation the views are directly used for setting rendering properties and adding graphical structures to be displayed.

The rendering of graphics is possible after mapping a window and activating the view. The direct setting of properties obsoletes the usage of intermediate structures with display parameter, such as `Visual3d_ContextView`, actually the whole package `Visual3d` becomes redundant.

Collection of all high-level API methods of application views in V3d package.

`Visual3d` layer has been eliminated. All its methods that could be previously used by the application are now exposed and should be accessed on the level of `V3d` entities.

New class `Graphic3d_CView` has been introduced as the base class for render views replacing `Visual3d_View` class. It allows specializing concrete instances of the class by a graphical driver. All methods for view rendering have been also moved into the interface of `Graphic3d_CView`. The corresponding methods have been removed from the graphical driver interface.

Elimination of Visual3d package

The logic of managing display of structures has been put from `Visual3d_ViewManager` into `Graphic3d_StructureManager`.

Visual3d_View class has been removed and the logics of managing computed structures transferred to the base layer of `Graphi3d_CView`.

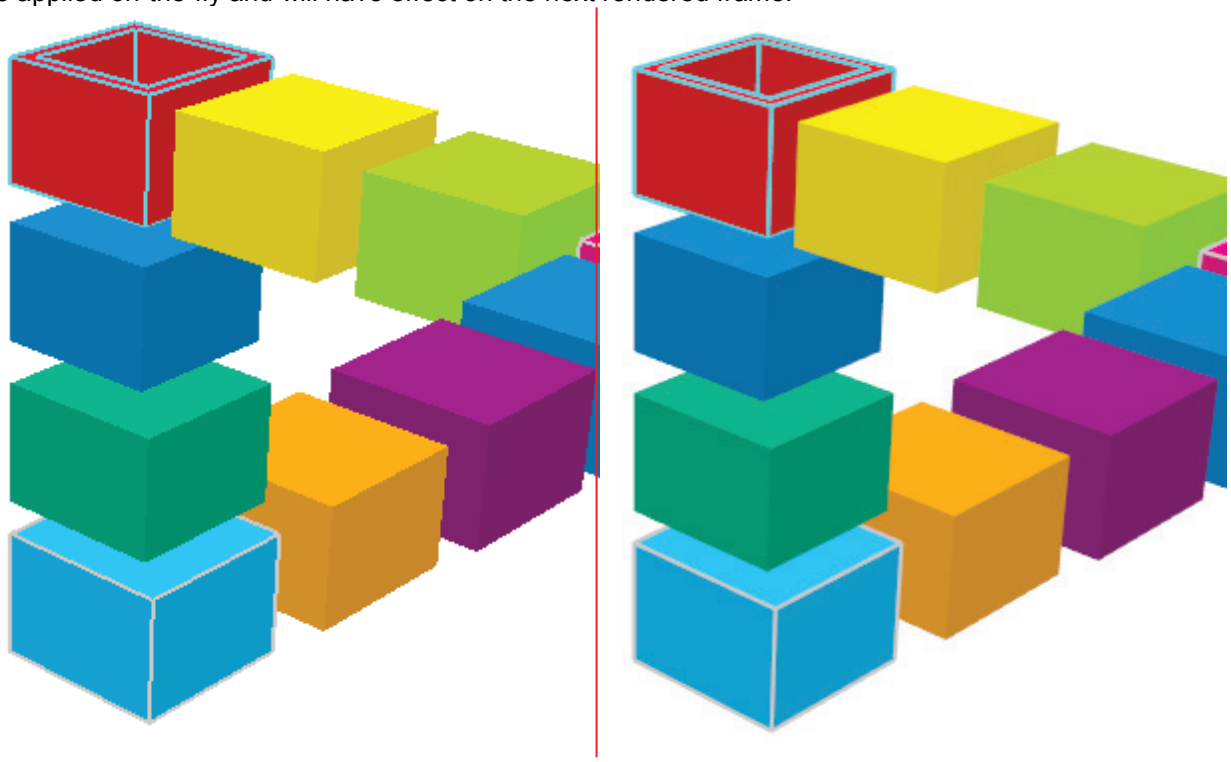
All intermediate structures for storing view parameters e.g. `Visual3d_ContextView` have also been removed. All settings are kept by the instances of `Graphic3d_CView`.

Intermediate class `Visual3d_Light` has been removed; all light properties are still stored in `Graphic3d_CLight` structure, which is directly accessed by instances of `V3d_Light` classes.
All required enumerations have been moved into `Graphic3d` package

Multisampling antialiasing

OCCT 3D Viewer has been extended with an option defining antialiasing options. The application just needs to assign the number of samples for MSAA buffer to `Graphic3d_RenderingParams::NbMsaaSamples` of view rendering parameters returned by `V3d_View::ChangeRenderingParams()` method.

The number of samples should be power-of-two and depends on the graphics hardware – the usual upper limit (the highest quality) is 8 samples, zero (default value) means that MSAA is turned off. This option can be applied on-the-fly and will have effect on the next rendered frame.



Without MSAA (left side) and with MSAA 8 (right side)

Deprecation of non-programmable rendering pipeline

In this release OCCT moves towards usage of a programmable rendering pipeline (GLSL programs) which now covers all major functionality.

The obsolete fixed function pipeline is still used by default on desktop platforms for compatibility, but is deprecated since OCCT 7.0.0 and will be removed in the future. This also means that baseline requirement for OCCT 3D viewer on desktop platforms is now OpenGL 3.3 or later.

It is recommended to disable the deprecated functionality by setting `OpenGl_Caps::ffpEnable` flag to `FALSE` within `OpenGl_GraphicDriver::ChangeOptions()` before creating the viewer.

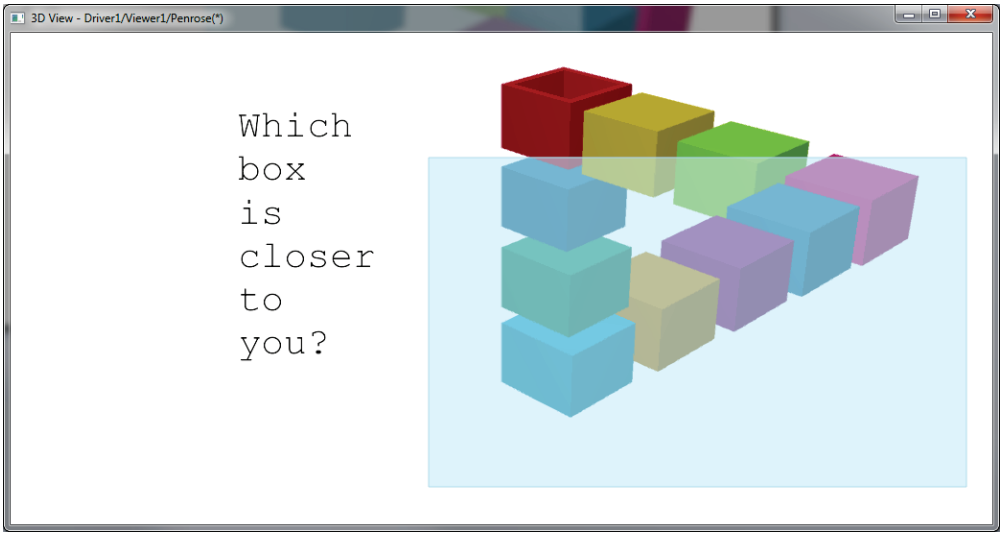
Note that the use of 3D viewer requires defining the environment variable `CSF_ShadersDirectory`, which points at the directory where shaders are located (`src/Shaders` in OCCT sources). If the shaders are not found, the 3D view will fail to initialize.

Rubber-band

The list of standard presentation objects has been extended with class `AIS_RubberBand` for displaying a rubber band (hollow or filled) during rectangular or polygonal selection.

This new presentation relies on 2D on-screen rendering capabilities and redesigned immediate mode rendering features introduced in OCCT 6.9.0 release. It is possible now to draw rubber band using only standard OCCT interface and with minimal latency (without redrawing entire 3D viewer content).

The new object is now used in Draw Harness and standard MFC samples replacing platform-dependent drawing APIs (which previously caused artifacts).

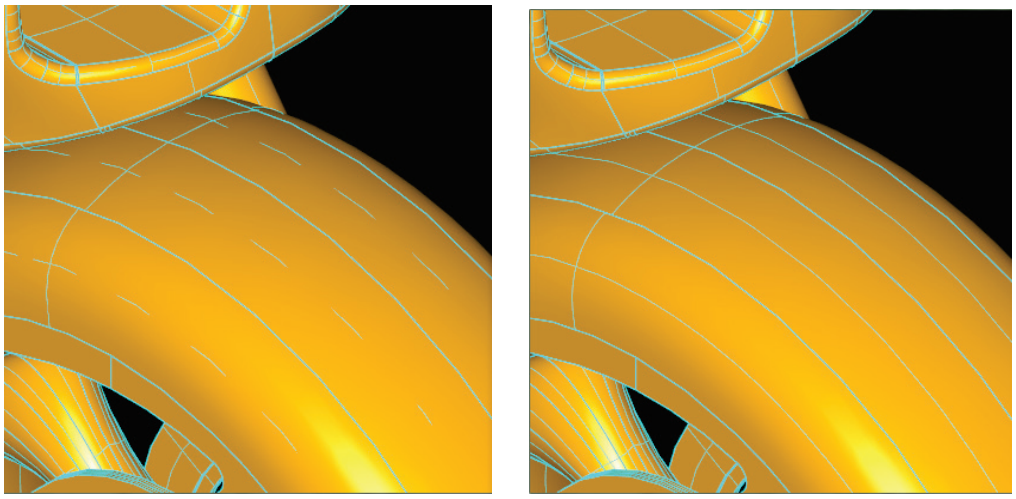


Rubber-band selection in Draw Harness

Isolines on triangulation

Traditional wireframe presentation of the geometry has been improved with the new option – mapping isolines onto triangulation. This new mode eliminates visual artifacts (vanishing and intersecting lines) usually caused by inconsistent discretization of isoline curves on surface and triangulation of the surface.

This option is disabled by default and controlled by `Prs3d_Drawer::SetIsoOnTriangulation()` method.



Conventional isolines (left) and isolines on triangulation (right)

Modifications

Foundation Classes

24023 26377 26457 26549 27014 27111	*Summary:* Revamp the OCCT Handle. The implementation of shared pointer used by OCCT (Handle) has been replaced by a modern solution based on C++ templates. See the details in New Features section. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
22325	*Summary:* Patch to fix a build failure on `GNU/kFreeBSD.` `FreeBSD_kernel` system type is now recognized in `OSD_Path`.
24537	*Summary:* GCC compiler warnings in byte order reversion code. Byte order inversion functionality is now provided using unions in class `FSD_BinaryFile`. Inversion functions in class `FSD_FileHeader` have been removed.
24780	*Summary:* Exceptions definition - move generated implementation from cxx to hxx. OCCT exceptions are now completely defined by macro `DEFINE_STANDARD_EXCEPTION` in the header file, no C++ code is necessary. Macro `IMPLEMENT_STANDARD_EXCEPTION` becomes obsolete; it is still defined as empty for compatibility.
24806 24947 26551 26913 26936 27016	*Summary:* Redesign of OCCT type system. OCCT type system has been redesigned. See the details in New Features section. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
24836	*Summary:* Stack overflow when raising exception in low memory condition `Standard_OutOfMemory` exception has been refactored to avoid memory allocations (which will likely fail) when it is raised: ▪ method `NewInstance()` returns a static instance (singleton); ▪ method `Raise()` raises a copy of that singleton, resetting its message string; ▪ message string is stored as a field, not allocated dynamically (thus the maximum message length is limited by buffer size). Class `Standard_Failure` has been slightly revised: method `Destroy()` merged to destructor, methods `Get/SetMessageString()` made virtual.
25574	*Summary:* `gp_YawPitchRoll` Euler Angle computation gives wrong results Conversion of `gp_Quaternion` to and from intrinsic Tait-Bryan angles is now done in correct order. For example, `gp_YawPitchRoll` equivalent to `gp_Intrinsic_ZYX` defines intrinsic rotations around Z, then Y, then X, not in reverse order as before.

26360	*Summary:* Useless global functions `IsSimilar()` in `Standard`. Definitions of global functions `IsSimailar()` for primitive types (such as `Address`, `Integer`, `Real`, `Character`, `CString`, etc.) have been replaced by `IsEqual()` and consequently removed from package `Standard`.
26364	*Summary:* `TKMath` - Optimize BVH binned algorithm. BVH binned builders (`BVH_BinnedBuilder` and `BVH_SpatialMedianBuilder`) have been optimized to improve BVH construction performance.
26381	*Summary:* `OSD_File` - close file on destruction. A destructor unlocking and closing an open file, has been implemented for `OSD_File`.
26514	*Summary:* `OSD_Path` cannot work with French symbols in file name. The restriction that a path should contain only basic ASCII symbols has been removed in `OSD_Path`. Any symbols defined in UTF-8 encoding are now possible.
26890	*Summary:* `TKernel` - define `OSD_OpenStream` for `std::ifstream`. New function `OSD_OpenFile::OSD_OpenStream` has been implemented to open file stream by file path defined in UTF-8 encoding.
27208	*Summary:* `Show` method in `Message_ProgressIndicator::NewScope` should not be commented out. `Show` method has been restored in `Message_ProgressIndicator::NewScope` to display the scope name in the progress indicator.
27281	*Summary:* Some classes in `GCPnts` are not const-correct. The keyword `const` has been added to method parameters in the classes from package `GCPnts`.

Application Framework

23465	*Summary:* Weird `InsertBefore`, `InsertAfter` and `Remove` methods in `TDataStd` lists. The interfaces of methods `Remove`, `InsertAfter` and `InsertBefore` in the List-classes from package `TdataStd` now provide access to the elements by an index.
23741	*Summary:* Research and remove `CSFDB` support from OCCT if it is necessary. Redundant `*.csfbd` files and `CSFDB` definition have been removed from the sample project.

24927	*Summary:* Getting rid of old “Persistent” functionality. The obsolete and unused Persistent functionality has been removed from OCCT. ▪ Standard persistence packages `ShapeSchema`, `StdLSchema`, `StdSchema`, `XCAFSchema` and all persistence-specific toolkits have been removed. ▪ OCCT custom formats `CSFDB`, `MDTV-Standard` and `MDTV-XCAF` are now deprecated; `Brep`, `BinOcaf` and `BinXCAF` should be used instead. ▪ The class `Standard_Storable` and all its uses have been removed. See the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
26005	*Summary:* Problem with transient `Tfunction_Logbook` `Tfunction_Logbook` has become an OCAF attribute. It keeps the modifications in OCAF tree and the data may be easily accessed through this attribute. Also, its data is updated on Undo/Redo operations.
26229 27077	*Summary:* Add the possibility in OCAF to open/save a document from/to a stream object. `TdocStd_Application` class now allows open/save a document in `XmlOcaf` and `BinOcaf` format from/to a standard stream object. The additional argument `-stream` has been added in Draw commands `Open` and `SaveAs` to turn on using stream functionality. Unused class `FSD_Archive` and its siblings have been removed from MFC samples.
26290	*Summary:* It is necessary to separate visualization part from TKCAF. TKCAF has been separated into two parts: ▪ The sources independent from visualization remain in TKCAF. ▪ Visualization dependencies have been moved to the new toolkit TKVCAF. See the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
26415	*Summary:* <`Tfunction_GraphNode`> XML should not break line in the middle of text. In method `XmlMFunction_GraphNodeDriver::Paste` the break between “previous” and “next” ids has been replaced by additional spaces to facilitate visual separation of two sub-lists.
26428	*Summary:* `Tnaming_Selector::Solve` crash for empty named shape. New function `Dnaming_ImportShape::CheckNSIter` checks if the input shape is new or old. `Selector` has been corrected to handle empty shapes properly.
26961	*Summary:* Recover possibility to read files in old persistence format. The possibility to read files in old persistent format is supported using toolkits `TKStd`, `TKStdL` and `TKShape`. Obsolete interfaces have been removed from classes in `PCDM` and `Storage`.

Modeling Data

23620	*Summary:* Make Bezier curve / surface evaluation thread-safe. Cache for Bezier curves has been removed from `Geom_BezierCurve`, `Geom2d_BezierCurve` and `Geom_BezierSurface` into `GeomAdaptor_Curve`, `Geom2dAdaptor_Curve` and `GeomAdaptor_Surface`.
24682 26949 27048 27107	*Summary:* Move out B-spline cache from curves and surfaces to dedicated classes `BsplCLib_Cache` and `BsplSLib_Cache`. B-spline cache has been separated into classes `BsplCLib_Cache` for 2D and 3D curves and `BsplSLib_Cache` for surfaces. The cache is used in the adaptor classes `Geom2dAdaptor_Curve`, `GeomAdaptor_Curve` and `GeomAdaptor_Surface` when the curve or surface is a B-spline. The algorithms have been changed to use adaptors for B-spline calculations instead of direct use of evaluation methods provided by curves or surfaces.
25342 26252 26838 26914	*Summary:* `GeomAdaptor_Surface` should use the inner adaptor to calculate values of complex surfaces. New package `GeomEvaluator` provides interfaces for calculation of values and derivatives for offset curves and surfaces including offset surfaces, surfaces of revolution and surfaces of extrusion. Its classes work with adaptors, curves and surfaces. Additionally, `Adaptor3d_SurfaceOfLinearExtrusion` and `GeomAdaptor_SurfaceOfRevolution` have been moved to `GeomAdaptor` and calculation of their values and derivatives has been unified. Obsolete namespace `CSLib_Offset` has been removed.
26255	*Summary:* Rename `Adaptor3d_OffsetCurve` into `Adaptor2d_OffsetCurve` reflecting its actual purpose. `Adaptor3d_OffsetCurve` has been renamed to `Adaptor2d_OffsetCurve` because it makes only 2d offsets. Redundant class `Geom2dGcc_CurveToolGeo` has been removed.
26526	*Summary:* `BrepTools_WireExplorer` cannot explore all edges of a closed wire. It has been explained in the comments to `BrepTools_WireExplorer` class that it works correctly only with a valid wire without any defects.
26755	*Summary:* Use of reference to destroyed temporary object in `Adaptor3d_SurfaceOfRevolution`. The method `Adaptor3d_SurfaceOfRevolution::GetType()` now makes a copy of temporary object for its further use.
27021	*Summary:* `TopExp::Vertices` performance optimization. Method `TopExp::Vertices` has been optimized for sequential calls. Performance regression in `Brep_Tool::Curve` has been fixed.
27059	*Summary:* Point->Curve Projection/Extrema fails [OCCT 7 only]. It is now checked in `Extrema_GextPC::Perform` method if the first derivative of objective function is small enough before the search for extrema is started.

Modeling Algorithms

21564 26530	*Summary:* Intersection of two planar faces produces curve with too many poles. `IntTools_FaceFace` now deletes excess points in intersection line generated at the first step of the algorithm (the points are deleted if the distance between them or chordal deviation is less than requested precision). In most cases this leads to generation of intersection curves with much less poles.
24357	*Summary:* `BRepBuilderAPI_Sewing` returns result with too high tolerance. Calculation of maximal deviation in method `Approx_SameParameter::Build()` considers that the projection is successful only if the projected point falls within the current interval of parameters (if 2D and 3D curves are not `same parameter`). Method `Approx_SameParameter::ProjectPointOnCurve()` takes tolerance into account. `BRepAlgoAPI_Sewing` catches the exception and properly computes edge tolerance if `same parameter` changes after the check with `BrepCheck_Analyzer`.
24890	*Summary:* Result of non-uniform scaling is invalid The functions `CorrectVertexTol`, `ModifiedShape` and `Modified` from `BRepBuilderAPI_NurbsConvert` class and `EvalAndUpdateTol` from `BrepTools` class have been implemented to improve tolerance evaluation in non-uniform scaling.
25709 25929 26472 26847	*Summary:* Make `Approx_ComputeLine` algorithm adaptive The algorithm of adaptive partition of `Wline` has been implemented in class `ApproxInt_Approx`. It chooses points for Bezier curve creation more accurately, to get a simpler output Bspline curve.
25813	*Summary:* Regression in Hidden Line Removal. A regression has been fixed in method `HLRBRep_Data::Update`.
25926 26837 27029	*Summary:* 3D offset in mode `Complete` with Join type `Intersection.` 3D offset algorithm has been extended to work in mode `Complete` with Join type `Intersection`. In this mode some faces of the original shape may have no mapping in the result. This extension is limited to work with planar faces only. Not all configurations of the initial shapes are yet supported. Thus, the possibility to produce an empty result in case of any invalidity (spikes, self-intersections, faces inversion) has been added. New option `RemoveIntEdges` in `BrepOffset_MakeOffset` allows removing INTERNAL edges from the faces of the result of Offset operation. By default the edges are kept in the result. To remove them, the corresponding flag should be set to `TRUE` when initializing the Offset algorithm.

26132	*Summary:* Invalid result of Boolean operation The procedures checking for Edge-Edge and Edge-Face coincidence have been added. These methods can be used instead of searching interferences between the corresponding sub-shapes. In most cases, new methods are more reliable and faster than intersections. However, their use should be avoided when the edge does not coincide with edge/face of another argument evidently (e.g. if edge vertices are not in another edge/face). The interface of both `IntTools_EdgeFace` and `IntTools_EdgeEdge` has been changed (some fields and methods added/deleted).
26184 26395 26593	*Summary:* `GeomAPI_ExtremaCurveCurve` hangs on parallel b-spline curves. Class `math_GlobOptMin` has been improved to use fast algorithm to filter out coincident points in case of enormous number of solutions. The performance of the algorithm has dramatically increased for the case of nearly parallel non-analytical curves.
26244	*Summary:* Destructive results of simplification with DRAW command `unifysamedom` after intersection of two complex models. Class `ShapeUpgrade_UnifySameDomain` has been improved to properly handle closed edges.
26254	*Summary:* Inject `GeomAbs_OffsetCurve` into `GeomAbs_CurveType` enumeration. Additional enumeration value has been introduced for offset curve in `GeomAbs_CurveType` enumeration in the same way as for offset surfaces. This improvement simplifies the use of offset curves in `GeomAdaptor_Curve` and `Geom2dAdaptor_Curve`.
26288	*Summary:* Offset on faces with opposite orientation. The offset algorithm has been improved in method `BrepOffset_Offset::Init` to properly calculate normals if mirror transformations are associated with input objects.
26323	*Summary:* Tolerance computing unification. The method `IntTools_FaceFace::ComputeFastTol3d()` has been removed. Now, if the intersection result contains 3D and the corresponding 2D curves, the tolerance is computed using `BrepLib_CheckCurveOnSurface` algorithm, which checks `same parameter`. If the intersection result contains only 3D curves, the tolerance will be computed using `GeomAPI_ProjectPointOnSurf` algorithm, which projects a point of 3D-curve on the surface and finds the maximal distance.
26383	*Summary:* Incorrect tolerance computing in `IntTools_FaceFace::ComputeTolerance()` The implementation of method `IntTools_FaceFace::ComputeTolerance()` has been corrected.
26396	*Summary:* Taper API result differs run-to-run for identical inputs. All data maps where shape is used as a key have been replaced with indexed data maps in class `Draft_Modification`. Now the index is used for iteration through this map instead of shape-key, ensuring stable results.

26417 26431 26675 26752 26777 26896	*Summary:* Cannot cut a sphere from a cylinder The following improvements have been implemented in the processing of spheres and cylinders by Boolean operations: ▪ Normalization, i.e. transformation of source data coordinates to fit them in range [0, 1], which changes the curvature of the approximated line and sometimes impairs the approximation quality is now not used. ▪ The processing of singularity points on surface (especially sphere poles) has been improved in the algorithm of intersection line computation and approximation (computation of correct 2D- and 3D-tangency at the end of Bezier constraints). ▪ Now ranges of 3D- and 2D-curves are unified even if `BrepLib::SameParameter()` method fails. ▪ Interfaces of some methods in classes `AppDef_Compute`, `ApproxInt_MultiLine`, `BrepAlgo_BooleanOperations` and `IntPatch_Wline` have been changed. ▪ Some overloaded methods have been deleted from classes `ApproxInt_Approx` and `TopOpeBRepTool`. ▪ Draw command `2dintersect` now prints information about found segments.
26426 26427	*Summary:* Draft angle algorithm modifies input argument and the operation result has very large tolerance values. The Draft angle algorithm has been corrected to avoid modifying the tolerance of input shape. This yields the right result in some particular cases where earlier it was wrong.
26443	*Summary:* Offset surface hangs up. The algorithm of computation of D0 and D1 values on offset surface has been simplified in class `Geom_OffsetSurface` to improve the performance.
26460	*Summary:* Implicit cast to `TopoDS_Shape` compilation error due to ambiguous conversion. `BRepBuilderAPI` package has been revised to avoid declaring methods `operator TopoDS_Shape` and `Shape` as `const`, which could cause a compilation error.
26470	*Summary:* `BrepFill_Evolved`: exception and invalid result. Tolerance analysis in some classes of `BrepFill` package has been improved to provide better recognition of analytical cases.
26525	*Summary:* Wrong result obtained by curve / surface intersection algorithm. Protection against floating point overflow caused by untrimmed parameters space has been implemented in method `IntCurveSurface_Inter::DoNewBounds`.
26560 27092	*Summary:* `BrepBndLib` builds too large bounding box in Z direction for planar spline edge. New method `BndLib_Add3dCurve::reduceSplineBox` computes a more accurate bounding box of 3D curves by taking into account the bounding box of poles of Bezier and B-spline curves. New method `Geom_BezierCurve::Poles()` provides access to Bezier poles. `MaxTolerance` function has been added in `Brep_Tool` class to compute the maximum tolerance of a sub-shape.

26565 27270	*Summary:* `Compsolid` becomes a compound of solids after cut. The requirements to argument types for different Boolean operations have changed: ▪ The types of arguments within the groups of Objects and Tools may be different for CUT, CUT21 and COMMON operations; ▪ For CUT operation the minimal dimension of Tools should not be less than the maximal dimension of Objects. The type of the result of Boolean operation has also changed: ▪ The result of Boolean Operations may contain shapes of different dimension, but the minimal dimension of the operation result is defined by the type of the operation and dimension of the arguments: ○ The minimal dimension of the result of COMMON operation will be equal to the minimal dimension of the arguments; ○ The minimal dimension of the result of CUT operation will be equal to the minimal dimension of the Objects; ○ The minimal dimension of the result of CUT21 operation will be equal to the minimal dimension of the Tools; ▪ For the arguments of collection types (WIRE, SHELL, COMPSOLID) the type will be passed into the result. For example, the result of COMMON operation between a Shell and a Wire will be a compound containing the Wire. Thus, the new version of the algorithm generates a structured result, depending on the structure of the input shapes. There is a specific case when the arguments of Boolean operations are containers with overlapping parts: ▪ The overlapping parts of arguments of collection type (WIRE, SHELL, COMPSOLID) passed into result will be repeated for each container from the input shapes containing such parts. ▪ The result of the operation Fuse for arguments of collection type (WIRE, SHELL, COMPSOLID) will contain the same number of containers as the arguments. The overlapping parts (EDGES/FACES/SOLIDS) will be shared among them. For example, the result of Fuse operation between two wires will be two wires sharing coinciding edges, if any. ▪ The result of the operation Common for arguments of collection type (WIRE, SHELL, COMPSOLID) will consist of the containers with the same overlapping parts. For example, the result of Common operation between two fully/partially overlapping wires will be two wires containing the same edges. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
26572	*Summary:* Error in `ShapeUpgrade_UnifySameDomain` algorithm: cannot merge two edges in a shape. Obsolete code has been removed from the method `ShapeUpgrade_UnifySameDomain::MergeEdges`.
26576 26684	*Summary:* Wrong result obtained by intersection algorithm. New function `DecomposeResult()` from class `IntPatch_ImpPrmIntersection` provides the algorithm of Restriction line decomposition. It checks if the source intersection (restriction) line should be split, computes and adds split points and creates a set of restriction lines from the source one. Then every line is approximated to a separate `Geom_Curve`.

26576 26684	New function `IsCoincide()` from the same class provides an algorithm checking for coincidences between Walking and Restriction line. This function replaces `IsIn2Dbox()`, thus providing more accurate check for coincidence.
26609	*Summary:* Wrong result obtained by solid classifier algorithm. The processing of "inverted" shapes (representing a cavity in the material) by solid classifier algorithm has been improved in methods `IntCurvesFace_Intersector::SurfaceType()` and `BrepClass3d_SolidExplorer::OtherSegment`.
26619 26796 27032	*Summary:* Tolerances of operands are modified using BOP. Boolean Operation algorithms can now work in 'protected' mode when modifications of tolerances of sub-shapes of arguments, necessary in some cases to ensure their proper interferences, do not propagate to input shapes. In this mode, new sub-shapes (vertices and edges) are created when the tolerance of a sub-shape needs to be amended. This new behavior is turned off by default. It can be turned on using two ways: ▪ by setting '`locking`' flag of the arguments; ▪ by calling method `SetNonDestructive(Standard_True)` of the API classes.
26621	*Summary:* Boolean Cut does not work on two solids. The function `IntPatch_PrmPrmIntersection::SeveralWlinesProcessing` now provides correct line extension.
26674	*Summary:* Performance regression in `BrepExtrema_DistShapeShape.` `BrepExtrema_DistShapeShape` now uses a single-solution mode for analysis of distances between curves (see `SingleSolutionFlag()` method in Extrema_ExtCC algorithm), thus it will report not more than one solution between each pair of edges (e.g. for parallel edges). See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
26681	*Summary:* `BRepPrimAPI_MakeRevol` creates a faulty shape. New method `BrepLib::UpdateInnerTolerances` checks tolerances of edges and vertices of a shape and updates them to satisfy `same parameter` condition.
26699	*Summary:* Wrong section curves. The algorithm of restriction line processing provided by class `IntTools_FaceFace` now checks if a line intersects true boundaries and enlarges the bounding box if a point on curve is inscribed in it. The algorithm of checking if Restriction line and Walking line coincide has been improved in the method `IntPatch_ImpPrmIntersection::Perform`. New function `IsTangentExtCheck` implemented in class `IntWalk_Iwalking` checks if the starting point of Walking line is a tangent point.

26717 26985 27159	*Summary:* Error in `IntPatch_PrmPrmIntersection`: change of local resolution leads to break of walking line. The method `IntWalk_Pwalking::TestDeflection` has been modified to take into account the local resolution of the chosen surface during the computation of the local step and the maximum step.
26718	*Summary:* Big tolerance value of the edge in the result of General Fuse operation. The distances between 2d and 3d curves (through surface) are now checked in method `BOPTools_AlgoTools2D::AttachExistingPCurve` before a 2d curve is attached to the edge to avoid huge tolerance increase.
26736	*Summary:* Errors in `BRepOffsetAPI_MakeOffset`: overlapping arcs are processed incorrect in mode `GeomAbs_Intersection.` The method `BrepFill_TrimEdgeTool::IntersectWith` now treats differently the case of multiple intersections of offset arcs and the case of extra-intersection of prolonged offset curves.
26746	*Summary:* Method `gp_Torus::Coefficients()` returns incorrect value. The calculation of torus coefficients in the absolute Cartesian coordinate system has been corrected in method `gp_Torus::Coefficients`.
26750	*Summary:* Method `gp_Vec2d::IsNormal()`returns `FALSE` if the angle between two vectors is equal to -90 degrees `(-M_PI/2 radian).` The method `gp_Vec2d::IsNormal` has been corrected to properly determine the normal in a specific case.
26757	*Summary:* Wrong history of a fillet. The method `ChFi3d_Builder::PerformIntersectionAtEnd` has been improved to provide the correct history of modifications in fillet construction algorithm.
26775	*Summary:* Bad result of section algorithm. The walking line computation method is now provided by a separate class `IntTools_WlineTool`. A set of methods (`IsDegeneratedZone`, `IsPointInDegeneratedZone`, etc.) has been implemented to determine if `Wline [ifprm, ilprm]` crosses the degenerated zone on each given surface. If yes, the surface with crossing in the degenerated zone is not taken into account during approximation.
26798	*Summary:* Boolean operations: Keep desired cells and boundaries in the result. The new class `BOPAlgo_CellsBuilder` allows adding or removing any part to (from) the result of Boolean fuse operation and removing any internal boundaries between those parts. Ultimately it allows performing arbitrary Boolean expressions on an arbitrary number of arguments. See the details in New Features section.

26841	*Summary:* Boolean operation `bsection` produces an invalid result on the attached cases. It is now checked in method `IntTools_FaceFace::MakeCurve()` if the intersection curve is bounded by faces domain to avoid increasing edge tolerance.
26871	*Summary:* Projecting a curve hangs inside `Approx_FitAndDivide2d`. The check of B-spline parametrization has been fixed in method `ShapeConstruct_ProjectCurveOnSurface::Perform`.
26884	*Summary:* Cylinder/Cylinder intersection algorithm throws an exception. It is now checked more carefully if cylinder axes are parallel in `IntAna_QuadQuadGeo`.
26897	*Summary:* `BRepBuilderAPI_Copy` does not copy polygons. The full mesh structure copying by `BRepBuilderAPI_Copy` with enabled flag `copyMesh` now includes copying of 3D polygons and polygons on surfaces. Previously they were omitted and the mesh had to be rebuilt.
26929	*Summary:* `Extrema_ECC` hang/crash. Comparator functor has been corrected in class `Extrema_GenExtCC` to return false in case of equal elements.
26955	*Summary:* Invalid result of General Fuse operation. The tolerance for point has been corrected in Solid Classifier algorithm provided by class `BrepCheck_ToolSolid`.
26967	*Summary*: `BrepFill_OffsetWire` should not copy plane if `Alt == 0.0`. In class `BrepFill_OffsetWire` the base plane is not copied anymore if the altitude parameter is equal to zero.
26980	*Summary:* Intersection part of Boolean algorithm spends much system time and system memory. A step of Walking-line has been increased as much as possible and is now computed by an iteratively adaptive algorithm (every iteration checks if the current step is too big/small and decreases/increases it).
27010	*Summary*: Wrong classification of the point relatively the solid. The algorithm of curve-face intersection has been improved in class `IntCurvesFace_Intersector` to take in account edge tolerance zones for more precise classification of intersection point state.
27035	*Summary*: General fuse algorithm loses face. New function `IsClosed` has been implemented in `GeomLib` class and used in BOP to check if the edge is a seam-edge on a closed surface or not.

27063	*Summary*: `BrepGProps` raises exception on edge with no curves. Method `BrepGProp::LinearProperties()` has been corrected to treat edges having zero length. This prevents exception on shapes that contain such edges, e.g. copy of a wire containing a degenerated edge. Method `Brep_Tool::IsGeometric()` has been optimized to avoid nested iteration for check of 3D curve for Null.
27065	*Summary*: `BRepOffsetAPI_MakePipe` misses definition of virtual method `Generated()`. Virtual method `Generated()` inherited from `BRepPrimAPI_MakeSweep` is overridden in class `BRepOffsetAPI_MakePipe`, providing information on shapes generated from the profile.
27066	*Summary*: `BrepFeat_MakeCylindricalHole::Perform()` hides overloaded virtual function. The base class member of `BrepFeat_MakeCylindricalHole` has been unhidden to avoid Clang warnings.
27085	*Summary*: `ShapeUpgrade_UnifySameDomain` very large performance difference for seemingly similar shapes. Performance of the algorithm `ShapeUpgrade_UnifySameDomain` has been improved by avoiding multiple calls of `ShapeBuild_ReShape::Apply()` method.
27100	*Summary*: Draw command `parameters` cannot compute parameter on the line. Calculations for analytical geometry have been removed from `GeomLib_Tool` algorithm, only use of `Extrema_<>` is preserved. Useless variable `MAXTOLERANCEGEOM` has been removed to allow placing a point at any distance from curve/surface.
27114	*Summary*: `DistShapeShape` does not find a solution edge-face. Extrema Curve / Surface algorithm implemented in class Extrema_ExtCS now performs more accurate search.
27131	*Summary*: `DistShapeShape` performance loss. The approximation of Lipchitz constant has been added in `Extrema_GenExtCC::Perform()`. The performance of global optimization algorithm `math_GlobOptMin` has been improved.
27133	*Summary*: Incorrect result of the normal projection algorithm Geometric tolerances are now used instead of topological ones in `ProjLib_CompProjectedCurve::Init()`. Protection from an invalid result has been added to restore the projection.
27134	*Summary*: Offset algorithm produces an invalid shape from a cone. The work of offset algorithm on conic faces has been improved in method `BrepOffset_MakeOffset::CorrectConicalFaces`.

27135	*Summary*: Incorrect result of the normal projection algorithm. A check for possible local traps has been added in class `ProjLib_CompProjectedCurve` to build a correct projection cache.
27162	*Summary*: Draw command `(2d)extrema` incorrectly represents underlying algorithm results. Handling of infinity solutions has been corrected for Curve / Curve case in `Extrema_ExtCC2d::Results`.
27175	*Summary*: Intersection algorithm with increased tolerance works incorrect with some shapes The algorithm checking for coincidence between Walking and Restriction line has been improved in `IntPatch_ImpPrmIntersection`.
27190	*Summary*: `IntPatch_ImpPrmIntersection` algorithm does not split intersection curve by the seam-edge of the quadric. Processing when `IntPatch_WLine/IntPatch_RLine` goes through the seam edge has been improved in `DecomposeResult(...)` function (see `IntPatch_ImpPrmIntersection.cxx`). Incorrect initialization of the last point of `IntPatch_WLine/IntPatch_RLine` has been eliminated. Earlier it was the reason of exception.
27207	*Summary*: New universal method extracting `HLRBRep_Algo` algorithm results. New method `HLRBRep_HLRToShape::CompoundOfEdges` returns a compound of edges of the specified type, visibility and space localization (3D/2D).
27273	*Summary*: The computation of linear properties on shared shapes is not correct. New flag `SkipShared` has been added in static methods `LinearProperties()`, `surfaceProperties()` and `volumeProperties()` from class `BRepGProp`. This flag defines if the second and next appearances of shared topology entities (edges, faces, shells) should be skipped during properties calculation.

Visualization

22016	*Summary:* Incorrect selection color after displaying a selected object. The problem with incorrect selection color after the display of a selected object has been fixed in method `AIS_InteractiveContext::Display()`.
23117 24776 26915 26792	*Summary*: inherit `OpenGl_View` from `Graphic3d_Cview`. OpenGL graphic rendering methods (the interface of `OpenGl_View`) have been exposed to the client code. See the details in New Features section.
24272	*Summary:* Provide basic text formatting routines for `Font_BrepFont` The method `Font_BrepFont::RenderText()` has been replaced by `Font_BrepTextBuilder::Perform()`, which now takes an optional formatter argument for text alignment.
24467 26732	*Summary:* `TKOpenGl` – add option to request Core profile 3.2+ using GLX. The following modifications have been introduced to support creation of core profile using GLX: ▪ `Aspect_Window` interface has been extended by new method `NativeFBConfig()` provided by implementations. ▪ `Xw_Window` now takes additional argument `GLXFBConfig`. ▪ `OpenGl_Window` does not implicitly create a child window when `Xvisual` for the passed window is incomplete in OpenGL context. This eliminates event-handling issues caused by unexpected window, but requires its proper creation for OpenGL usage.
25148	*Summary:* drop `TKNIS` toolkit. `TKNIS` toolkit has been removed because of obsolescence and incompleteness.
25162	*Summary:* `TKOpenGl` – drop `GLU` library dependency The fallback code for using non-power-of-two (NPOT) textures on old hardware has been dropped. The texture initialization will fail now on the hardware without proper support of OpenGL 2.0+. The fallback code for generating mipmaps for a 2D texture on old hardware has been dropped. The texture initialization will not create mipmaps with appropriate warning on the hardware without proper support of OpenGL 3.0+. `TKOpenGl` does not depend on deprecated GLU anymore.
25201 25347 26437	*Summary:* Implementing soft shadows and ambient occlusion in OCCT ray-tracing core. Additional rendering algorithm of OCCT ray-tracing engine allows handling global illumination effects (soft shadows, glossy reflections, color bleeding, diffuse inter-reflections, and caustics) and producing photorealistic output images. The implementation is based on path tracing that uses Monte-Carlo integration of rendering equation. The patch also introduces the extended material model that was smoothly integrated into existing OCCT material description classes.

25300	*Summary:* Build wireframe representation consistent with the shape's triangulation. `AIS_Shape` now supports additional algorithm for mapping isolines presentation onto triangulation. This option is controlled by flag `Prs3d_Drawer::IsoOnTriangulation()`, turned off by default. The following modifications have been introduced: ▪ Redundant presentation algorithms for shapes `StdPrs_WFShape` and `StdPrs_WFDeflectionShape` have been removed. ▪ `Prs3d_WFShape` has been renamed into `StdPrs_WFShape` and rewritten to use deflection for non-triangulated shapes. ▪ `StdPrs_ToolShadedShape` has been renamed into `StdPrs_ToolTriangulatedShape` (reused in `StdPrs_WFShape` and `StdPrs_ShadedShape`). ▪ New class `StdPrs_BndBox` allows drawing bounding box presentation. ▪ Option `-isoontriangulation` has been added to Draw command `vaspects`, which enables on-triangulation iso-line shape builder. ▪ The maximum UV parameter value of a drawer is now properly taken into account for iso-line calculation.
25305	*Summary:* `TKOpenGl` – support stipple line aspects within built-in GLSL programs. TKOpenGl has been improved to support stipple line aspect when using built-in GLSL programs instead of deprecated fixed-function rendering pipeline. The option `-setlinetype` has been added in Draw command `vaspects`.
25338 27039	*Summary:* MFC standard samples: 3D selection rectangle blinking. New interactive object `AIS_RubberBand` has been introduced for rendering rubber band directly within OCCT viewer. MFC samples and Draw Harness have been updated to use new presentation object within rectangular selection instead of platform-dependent window rendering
25508 25785 25789 26312 27058	*Summary:* Get rid of obsolete 2d layers implementation. Obsolete 2D layer rendering API (`Clayer2d`) has been completely dropped in favor of AIS supporting appropriate functionality since OCCT 6.9.0. Draw commands `voverlaytext` and `vlayerline` now work with the new implementation of 2d layers. `ColorScale` has been removed as a global property of `V3d_View` together with associated methods `V3d_View::ColorScaleDisplay()`, `V3d_View::ColorScaleErase()`, `V3d_View::ColorScaleIsDisplayed()` and `V3d_View::ColorScale()` as well as classes `V3d_ColorScale`, `V3d_ColorScaleLayerItem` and `Aspect_ColorScale`. New interactive object `AIS_ColorScale` providing the same configuration API as previously `Aspect_ColorScale` and `V3d_ColorScale` should be used instead and displayed as 2D presentation. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.

25549	*Summary:* Do not crash at the attempt to display Angle dimension between two parallel lines. New method `AIS_AngleDimension::GetNormalForMinAngle()` allows drawing arcs in accordance with the input normal for the minimum angle and checking correctness of input circle parameters. Method `AIS_AngleDimension::InitTwoEdgesAngle()` handles cases of 0 and Pi angle depending on lines in edges and end points.
26056	*Summary:* `AIS_LengthDimension` cannot build dimension for face-edge or edge-face. The method `AIS_LengthDimension::InitEdgeFaceLength()` has been corrected to support face-edge and edge-face cases.
26112 27123	*Summary:* `TKOpenGl` – clipping and capping is broken when FFP is disabled on Linux. Clipping and capping functionality now works properly within programmable pipeline (GLSL), when Fixed Function Pipeline is unavailable or turned off.
26149	*Summary:* Depth buffer should not be written within Z-layers without `Graphic3d_ZlayerDepthWrite` flag. The following modifications resolve inconsistencies in Depth-buffer clearing within Z-layers list: ▪ Outdated API methods `V3d_View::EnableDepthTest()`, `V3d_View::IsDepthTestEnabled()` `Visual3d_View::ZbufferIsActivated()`, `Visual3d_View::SetZBufferActivity()`, `Visual3d_View::EnableDepthTest()`, `Visual3d_View::IsDepthTestEnabled()`, `Graphic3d_GraphicDriver::SetDepthTestEnabled()` and `Graphic3d_GraphicDriver::IsDepthTestEnabled()` conflicting with Z-layers API have been removed. ▪ Z-buffer is activated by default. Its state can be managed only by Z-layer flags. ▪ In `OpenGl_Workspace::updateMaterial()` writing into Depth buffer is not activated without `Graphic3d_ZlayerDepthWrite` flag. ▪ Method `OpenGl_Workspace::UseDepthWrite()` has been added to track `glDepthMask()` state.
26272 26594 26596 26779 27137	*Summary:* Provide the possibility to activate selection modes without opening the local context. The following modifications provide the possibility to activate selection modes without opening the local context: ▪ Picked or selected objects are now highlighted via owners instead of interactive objects; ▪ Support methods for owners have been added to `AIS_InteractiveContext`; ▪ Dynamically highlighted owners are now displayed in immediate mode; ▪ New argument `–local` has been added to `vselmode` command to enable selection in the local context for testing deprecated functionality; ▪ Selection filters are now completely supported in `AIS_InteractiveContext`; The selected items have ceased to be differentiated into current (the interactive context) and selected (the local selection): all calls to “current” have been replaced by calls to “selected” in terms of future local context removal;

26272 26594 26596 26779 27137	▪ `AIS_InteractiveObject::mySelectionMode` has been removed – now each selectable object can define its own selection mode for "global" selection of the whole object. The whole object selection mode is 0 by default for all standard interactive objects; ▪ Support of drawing immediate structures has been added in different layers; The Local Context API is deprecated since this release – new applications should not use it. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
26292	*Summary:* Parallelize queue-based BVH builders (subclasses of `BVH_QueueBuilder`). BVH queue-based builders (`BVH_BinnedBuilder`, `BVH_SpatialMedianBuilder`, and `BVH_SweepPlaneBuilder`) have been parallelized. Note that parallelization is disabled by default (1 thread is used for building) and can be configured using `theNumOfThreads` argument in `BVH_QueueBuilder` constructor. In order to support parallel mode, the corresponding BVH primitive set should provide thread safe implementations of `BVH_PrimitiveSet` interface (such methods as `Swap`, `Box`, and `Center`). Otherwise, the results will be undefined (for that reason, parallelization is disabled by default). To the moment, parallel BVH construction is used in OCCT ray-tracing core. On a quad-core CPU the speed-up is about 300%.
26298 26790	*Summary:* `OpenGl_Text` – make font resolution configurable. New parameter `Resolution` has been added to `Graphic3d_RenderingParams` structure with default value 72 ppi (as in the previous OCCT release). New method `::ResolutionRatio()` has been added to `Graphic3d_RenderingParams` structure defining scale ratio based on `Graphic3d_RenderingParams::Resolution` property (relative to the default 72 ppi resolution). The resolution property is handled by `OpenGl_Context` to adjust the line width and by `OpenGl_Text` to scale text appropriately. Draw Harness command `vrenderparams` has been extended by option `-resolution` for setting pixel density.
26317	*Summary:* `AIS_LengthDimension::CheckPlane()` is incorrect. `AIS_LengthDimension::CheckPlane()` now checks if the plane normal and direction are parallel.
26343	*Summary:* Zoom persistent text with 3D orientation. It has become possible to define the orientation of axes for zoom persistent text to allow drawing it on a generic plane. The functionality is implemented using methods `Prs3d_Text::Draw()`, `Graphic3d_Group::Text()`, `OpenGl_Text::OpenGl_Text()`, etc.

26344 26375 26719 27136	*Summary:* Provide support of zoom persistent selection. The zoom persistent flag is now handled by the selection algorithm. Thus auxiliary presentation objects with fixed size (annotations, tools) can be detected as normal interactive objects. New `Graphic3d_TransformPers` structure defines parameters and algorithm methods, including: transformation of projection and world view matrices; computation of a model-world transformation of the persistent object and computation of a transformed bounding box of the persistent object. Transform persistence algorithm does not make any changes to model-world transformation of object (deals with projection and world view matrices only), thus making possible to employ local transformation in a usual way. ▪ BVH selection and efficient frustum culling have been implemented for transform persistent objects (pan, rotate, zoom, trihedron persistence only); ▪ Z-fitting algorithm has been implemented for world-view space transform persistent objects (rotate, zoom persistence only); ▪ Usage of transform persistence structures and utility classes has been rewritten: `Graphic3d_CtransPers` replaced by `Graphic3d_TransformPers`; `OpenGl_Utils` by `Graphic3d_TransformUtils`. ▪ New class `Graphic3d_WorldViewProjState` keeps track of projection and world view matrix changes for a camera. ▪ Method `Graphic3d_Camera::ModelViewState()` has been renamed to `::WorldViewState()` for consistency.
26348	*Summary:* `TKOpenGl` – eliminate invalid NULL checks for transformation matrix Method `Graphic3d_Cstructure::Transformation` now uses `Graphic3d_Mat4` instead of plain arrays. Duplicate field `OpenGl_Structure::myTransformation` as well as unused properties `Graphic3d_Cstructure::Composition` and `Graphic3d_Structure::Composition()` have been removed.
26361	*Summary:* Move `OpenGl_TextFormatter` to `Font_TextFormatter` for usage without OpenGL. The class `OpenGl_TextFormatter` has been split into `Font_TextFormatter` and `OpenGl_TextBuilder`. `Font_TextFormatter` formats text independently from `OpenGl`. `OpenGl_TextBuilder` generates a primitive array required for text rendering using `OpenGl_Font` instance.
26363	*Summary:* `TKOpenGl` – missing `RayTracing` shader files should be properly reported in Release mode. `OpenGl_View::ShaderSource::Load()` method prints error about missing files.
26392 26542	*Summary:* `TKD3Dhost` – provide straight-forward base for integration of `TKOpenGl` viewer into D3D-based application. New class `D3Dhost_GraphicDriver` (`TKD3Dhost`) provides smooth integration of `OpenGl_GraphicDriver` (`TKOpenGl`) into `Direct3D 9` applications. It relies on D3D/WGL interoperability layer provided with OpenGL extension `WGL_NV_DX_interop` by main graphics hardware vendors on Windows platform (Intel, AMD and NVIDIA). Within the new module, the rendering itself is performed using OpenGL API, but the result image is presented by Direct3D9 API without extra memory copying.

26404	*Summary:* Ray Tracing – use solid background color when gradient color is disabled. The ray-tracing shader implemented in `OpenGl_View::setUniformState()` now takes into account solid background settings.
26413	*Summary:* Pixel tolerance is overridden by selection sensitivity. The custom pixel tolerance from `vselprecision` is now added to the default primitive sensitivity in `SelectMgr_ViewerSelector`. This facilitates selecting edges on touch screen displays.
26421	*Summary:* Incorrect text rendering in raytracing mode. The color of primitives drawn by conventional rasterization operations (e.g. texture-based text) is now properly mixed within Ray-Tracing program.
26433	*Summary:* `AIS_Dimension` may attempt to modify the state of default Drawer shading aspect. Checks to allow modification of only own aspects have been added in `AIS_Dimension::DrawText()`.
26435	*Summary:* `V3d_View::ConvertToGrid` returns wrong coordinates for non-orthogonal projection. The method `V3d_View::ConvertToGrid` now provides correct coordinates.
26511	*Summary:* Build fails with VTK 6.3rc1 Obsolete typedef `vtkFloatingPointType` has been replaced by `double` in VIS and DRAW
26536	*Summary:* Ray-tracing engine – improve BVH traverse and fix texture support. 64-bit handles of bindless textures have been replaced in various `OpenGL` package classes by `uvec2` type for compatibility with AMD drivers.
26566	*Summary:* Incorrect highlight after selection of owners with auto-highlight disabled. The algorithm canceling the highlighting of previously selected owners has been fixed in method `AIS_InteractiveContext::SetSelected()`.
26571	*Summary:* `TKOpenGl` - write depth values within Ray Tracing program. View-projection matrix has been added to raytrace shaders as uniform to compute correct depth values for OpenGL. Additional depth buffer sampler has been added to `Display.fs` program for path tracing. It allows propagation of depth values from internal FBO to resulting FBO. The old approach of mixing of OpenGL and ray-tracing graphics is preserved for correct blending of transparent ray-traced objects with non-transparent OpenGL objects.
26599	*Summary:* `TKOpenGl` – gradient background should reset model transformation in Core profile. `OpenGl_View::DrawBackground()` now resets not only `WorldViewState`, but also `ModelWorldState`.

26617 26676	*Summary:* Ray Tracing – correct rendering if stereo pair. Unnecessary normalization of direction vector affecting the frustum's geometry is now avoided during the interpolation of ray direction vector for asymmetric frustum (stereo left/right eye) in method `OpenGl_View::updateCamera()`. The usage of projection type argument passed when rendering immediate graphics with stereo projection has been fixed in method `OpenGl_View::Redraw()`.
26625	*Summary:* Possible wrong use of `vtkSmartPointer<T>` in `IVtkVTK_ShapeData.cxx` The method `vtkSmartPointer<XXX>::New()` is now used instead of `XXX::New()` when initializing fields of `vtkSmartPointer<XXX>` types.
26658	*Summary:* Unexpected selection in the context using a selection filter. Method `AIS_LocalContext::ClearOutdatedSelection()` chooses `mylastindex` value from the list of filtered detected owners. The topmost detected owner will be highlighted if the current detected owner was cleated as outdated.
26711 26834	*Summary:* `TKOpenGl` – support creation of multisampling off-screen FBOs. FBOs with multisampling textures have been implemented as follows: ▪ New method `OpenGl_Texture::Init2Dmultisample()` initializes multi-sampled texture. ▪ The option `NbMsaaSamples` defining MSAA samples number has been added to `Graphic3d_RenderingParams`. Ray Tracing continues using FBO without MSAA, however, it is possible to combine MSAA for rasterization and FSAA for RayTracing. ▪ `OpenGl_FrameBuffer` constructor has ceased to take arguments. ▪ The method `OpenGl_FrameBuffer::Init()` has been extended with mandatory parameters defining Color and Depth attachment formats and an optional parameter defining the number of MSAA parameters.
26721	*Summary:* Selection highlight of selected face is broken. Selection functionality has been fixed to work correctly with highlight.
26733	*Summary:* deviation angle cannot be set by `AIS_InteractiveContext::SetDeviationAngle()` due to misprint The method `AIS_InteractiveContext::SetDeviationCoefficient()` setting the deviation angle has been fixed.
26754	*Summary:* Provide API to display `AIS_Trihedron` presentation without axes labels. New API allows displaying `AIS_Trihedron` presentation without axis labels: ▪ New option `ToDrawLabels` has been added to `Prs3d_DatumAspect`, by default it is set to true ▪ `AIS_Trihedron` and `AIS_Axis` have been modified to draw labels only if `ToDrawLabels` option in the datum aspect is enabled.

26765 26710	*Summary:* Drop `TKVoxel` toolkit. All functionality related to voxels and provided by `TKVoxel` toolkit has been removed as obsolete by design and data structure definitions. A similar functionality can be provided by the algorithms based on Volume Rendering.
26768	*Summary:* `Graphic3d_Camera::ZfitAll()` – define method estimating `Zrange` without assigning it. Methods `V3d_View::Redraw()` and `V3d_View::RedrawImmediate()` have been declared virtual to allow customization.
26808	*Summary:* `TKOpenGl` – specify GLSL 120 for point sprites program. GLSL 120 has been specified for point sprites program in `OpenGl_ShaderManager` for compatibility with Mesa 3D OpenGL drivers.
26821	*Summary:* Default rendering parameters for interactive context. Default Rendering Parameters are now defined within `V3d_Viewer` and used by new instances of `V3d_View`.
26844	*Summary:* `OpenGl_Flipper` – fix issues within Core Profile and OpenGL ES. Outdated code has been removed from `OpenGl_Flipper::Render()` method.
26870	*Summary:* Deactivated selections are not updated after object re-computation. The method `SelectMgr_SelectionManager::RecomputeSelection()` has been fixed to update status to full for all selections and switch it to none only if the selection has been actually recomputed.
26891	*Summary:* `TKOpenGl` – define more texture types within `OpenGl_TextureFormatSelector`. Texture types `Glbyte`, `Glshort`, `Gluint` and `Glint` have been implemented in `OpenGl_TextureFormatSelector`.
26905	*Summary:* Cosmetic fixes in selection methods of `AIS_InteractiveContext` Selection mode has been corrected in `AIS_InteractiveContext::AddOrRemoveSelected()`. `AIS_InteractiveContext::IsSelected()` now returns object state instead of the global status.
26940	*Summary:* `TKOpenGl` – capping plane should be applied to connected structures The clipping plane is now properly handled when rendering connected structures on `TKOpenGl` level.
26945	*Summary:* Selection does not work after closing one of local contexts in stack New function AIS_LocalContext::RestoreActivatedModes() allows restoring selection of the local context. Activated standard modes are now added to the local status.

26959	*Summary:* Cannot select edge of the shape Polygonal representation of the edge is now used in `StdSelect_BrepSelectionTool` for sensitive entity computation regardless of the requested deflection.
26960	*Summary:* `TKOpenGl` – update transformation of dynamically highlighted presentation. New method `PrsMgr_PresentationManager::UpdateHighlightTrsf()` provides immediate update of transformation of highlighted presentation. Interfaces for immediate transformation update of the corresponding presentations have been added to entity owner classes.
26969	*Summary:* Support custom vertex attributes in GLSL program. The API for defining custom vertex attributes has been introduced in `Graphic3d_ShaderProgram::SetVertexAttributes()`.
26973	*Summary:* Selection of entities hidden by clipping planes is broken. Selection algorithm has been corrected to take into account clipping planes properly.
26975	*Summary:* `TKOpenGl` – handle triangle strips correctly within Ray-Tracing core. Triangle strips are now converted correctly by ray tracing uploader `OpenGl_View::addRaytraceTriangleStripArray()`.
26995	*Summary:* `TKXCAF` – do not reset custom material within `XCAFPrs_AISObject::Compute()`. `XCAFPrs_AISObject::XCAFPrs_AISObject` now defines default plastic material for proper color reproduction.
27060	*Summary:* Visualization issue with `TopoDS_Vertex` after call of `AIS_Shape::SetColor()`. Method `AIS_Shape::setColor()` now defines `Aspect_TOM_PLUS` point aspect in sync with `Prs3d_Drawer::PointAspect()` when `Prs3d_Drawer` has no Link.
27083	*Summary:* Ray Tracing – shape with visible face boundaries disappears after turning the ray-tracing on. The default state of aspects is now restored to prevent backface culling which is not yet supported by ray-tracing.
27172	*Summary:* Avoid signed integer overflow within `Graphic3d_ArrayOfPrimitives`. Signed integer overflow within `Graphic3d_ArrayOfPrimitives` is now avoided.
27180	*Summary:* Improve selection logic of `MeshVS_Mesh`. The performance of method `MeshVS_Mesh::ComputeSelection()` has been optimized by avoiding the creation of many independent sensitive entities for each mesh element or node. Now it creates a single sensitive entity. `MeshVS_SensitiveQuad` and `Select3D_SensitiveTriangle` are now used instead of `Select3D_SensitiveFace` for local selection to reduce memory consumption.

27286	*Summary:* TKOpenGl - avoid using light index within built-in GLSL programs for simplest configuration. OpenGl_ShaderManager::stdComputeLighting() handles single directional light specifically for compatibility with broken OpenGL ES drivers.

Data Exchange

24595	*Summary:* STEP import missing surfaces. The method StepToTopoDS_TranslateFace::Init has been modified to create natural bounds for a face based on the spherical or Bspline surface and having only one bound represented by a Vertex loop.
25441	*Summary:* XCAFDoc_ShapeTool::UpdateAssembly() does not update the back-references. Previously the method XCAFDoc_ShapeTool::UpdateAssembly() rebuilt the shape of assembly, however, it did not follow the back-references, i.e. the users of the assembly. Now this method checks back-references in the bottom-up direction to ensure the shape data consistency in an XCAF document. Consequently, all methods that call UpdateAssembly() have been reviewed, e.g. duplicated code has been removed from SetShape().
25522	Summary: GCC for Android cannot compile lex.step.c in Release mode. An updated version of lexical scanner lex.step.c is now used in OCCT.
26216	*Summary:* Convert a compound to assembly. New method XCAFDoc_Editor::Expand converts a compound (or all compounds contained in a document) into assembly. In Draw this functionality is provided by command Xexpand
26338	*Summary:* STL export (especially binary) needs a lot of time if epy selected export path is not local. Method StlAPI_Writer::Write() has been reimplemented to write triangulation directly, without conversion to StlMesh_Mesh. New DRAW command tessellate has been added to rapidly generate triangulation of prescribed size on surface. Command tricheck has been protected to deal correctly with triangulation without UV data.
26371 26720 26947	*Summary:* Implementation of new entities for GD&T. New entities required for support of geometric dimensioning and tolerances (GD&T) data in STEP AP242, have been implemented according to recommended practices from CAX-IF.

26451	*Summary:* Crash importing STEP file. Null check has been added in methods `StepShape_OrientedEdge::EdgeEnd()`, `StepShape_OrientedEdge::EdgeStart()` and `StepVisual_FillAreaStyle::FillStylesValue` to avoid crash.
26500	*Summary:* Add the possibility to save only part of shapes from XCAF document in IGES/STEP writers. New methods `Transfer()` in `STEPCAFControl_Writer` and `IGESCAFControl_Writer` provide the possibility to transfer only the part of assembly defined by the specified labels (shapes) in IGES/STEP. If the specified label is a component of a high-level assembly then this assembly is saved in a document with the specified component. In other cases only the part of the document starting from the specified label is saved.
26508	*Summary:* Add the possibility to get label from `XCAFPrs_AISObject`. The possibility to get label from `XCAFPrs_AISObject` has been implemented to determine quickly and precisely the label (and also the shape) in XCAF document, which is associated with a given presentation.
26715	*Summary:* Problems in reading STEP short names in complex entities. New function `StepData_StepReaderData::NamedForComplex` has been implemented to find the next part of a complex entity by its full or short name. Several short names have been added.
26751	*Summary:* STEP file produced from XDE document cannot be read back. The method `XtData_Field::Read` has been protected against dummy symbols in names, so that STEP translator correctly counts the number of parameters in `PRODUCT` entity.
26762	*Summary:* Static parameter `read.scale.unit` is not used. Unused parameter `read.scale.unit` has been removed from `XSAlgo.cxx`.
26859	*Summary:* Export of GDT from XCAF to STEP. Export of dimensions, geometric tolerances and datums has been implemented in STEP according to AP242. Missing STEP entities have been added. In XCAF, new attribute `DatumTargetNumber` has been added to `Datum` object.
26922	*Summary:* Huge performance issue writing data to the output stream. Throughout OCCT code `std::endl` has been replaced by '\n' where data are written to a file using `C++` streams. This significantly improves output performance, especially for VRML output.

26931 26989	*Summary:* [Regression in 6.9.0] Exporting a face throws an exception. The method `GeomToIGES_GeomSurface::TransferSurface` writing periodic BSpline surfaces to IGES now sets new origin for periodic BSpline surfaces for synchronization of pcurve ranges. The bounds of a face are fixed if its length (in U or V) is more than period. The BSpline curve/surface segmentation now throws an exception if the segment length is more than period. `*Raise_if` macros have been replaced with unconditional exceptions in classes `Geom2d_BSplineCurve`, `Geom_BSplineCurve` and `Geom_BSplineSurface`, wherever such replacement does not affect the performance.
26951	*Summary:* Incorrect conversion of miles into millimetres during export to STEP. The following coefficients for conversion between metric and non-metric units have been corrected: ▪ miles to mm set to 1609344 in `UnitsMethods::GetLengthFactorValue();` ▪ inches to mm set to exactly 25.4 mm in `UnitsAPI/Units.dat;` ▪ nautical mile set to exactly 1852 m DRAW command `unit` outputs to Tcl instead of cout.
27047	*Summary:* Eliminate useless polymorphic methods `Init()`. Methods `Init()` in STEP data classes have been made non-virtual, redundant variants just calling the same method of the base class are removed.

Draw

22632	*Summary:* Provide logarithmic scale for `Aspect_ColorScale` class. New option –logarithmic of Draw command `vcolorscale` allows changing color scale labels to logarithmic values correspondingly to the min/ max range and the number of intervals of the color scale. The call of command `vcolorscale` without arguments is now avoided.
25777	*Summary:* `ViewerTest` – make commands defining standard views to match their names. The implementation of `ViewerTest_ViewerCommands` has been corrected. Old `vright` equals new `vfront`, old `vfront` equals new `vright`, old `vleft` equals new `vback`, old `vback` equals new `vleft`.
26235	*Summary:* Command `tolmax` works wrong. New command `checkmaxtol` has replaced the obsolete command `tolmax` and is used now in test cases to check for the maximum tolerance.
26489	*Summary:* The class `ShapeUpgrade_UnifySameDomain` provides the results that are wrong or difficult to explain. Draw command `unifysamedom` now tries to merge all possible sub-sequences in the sequence of edges.

26490	*Summary:* Implement Draw commands `voverlaytext` and `vlayerline` using `AIS_InteractiveObject` class. Draw command `voverlaytext` has been removed and its functionality transferred to command `vdrawtext`.
26726	*Summary:* ViewerTest – `AIS_InteractiveContext::EraseSelected()` lacks test case. Excessive viewer update is now avoided in command `ViewerTest::Erase()`.
26855	*Summary:* Draw commands to debug Boolean Operations Algorithm. The following commands for debug of Boolean operations have been added in Draw: ▪ `bopds` – Shows the shapes from DS. ▪ `bopiterator` – Shows the pairs of interfered shapes. ▪ `Bopinterf` – Shows interferences of given type. ▪ `Bopnews` – Shows the newly created shapes ▪ `Bopwho` – Shows where the new shape was created ▪ `Bopindex` – Gets the index of the shape in the DS. ▪ `Bopsd` – Gets the Same domain shape. ▪ `Bopsc` – Shows the section curves. ▪ `Boppb` – Shows information about pave blocks. ▪ `Bopcb` – Shows information about common blocks. ▪ `Bopsp` – Shows the splits of edges. ▪ `Bopfon` – Shows ON information for the face. ▪ `Bopfin` – Shows IN information for the face. ▪ `Bopfsc` – Shows SC information for the face. ▪ `Bopfav` – Shows information about alone vertices for the face. ▪ `Bopimage` – Shows split parts of the shape ▪ `Boporigin` – Shows the original shape for the shape. ▪ `Bopfsd` – Shows SD faces for the face: ▪ `Bopbsolid` – Builds solids from set of shared faces ▪ `Bopbface` – Splits the face by set of shared edges.
26984	*Summary:* `ViewerTest` – preserve local transformation of presentation within command `vtexture`. The local transformation of object is now preserved after applying command `vtexture`.
27045	*Summary:* Commands `firsthole & holend` work incorrectly. Commands `firsthole`, `holend` and `hole` now work correctly on planar faces (creation of duplicate holes is avoided). The corresponding changes have been introduced in class `BRepFeat_MakeCylindricalHole`.
27293	*Summary:* Add debug function to save a list of shapes into a compound New function `DBRep_SetComp` allows saving a list of shapes into a draw variable as a compound.

Mesh

26384	*Summary:* Add explicit check for null magnitude instead of catching of exception in `BRepMesh_FastDiscretFace::control()` Method `BRepMesh_FastDiscretFace::control()` now checks normals for null magnitude using `gp::Resolution()`.
26407	*Summary:* `BRepMesh_Delaun` should not take into account frontier edges on first pass of algorithm The procedure of insertion of new vertices has been corrected in method `BRepMesh_Delaun::createTrianglesOnNewVertices` to remove all triangles shot by a point even if they contain frontier edges. This is done to avoid gluing free edges with the frontier. In function `UpdateBndBox` from class `BRepMesh_Delaun` bounding boxes have been enlarged by `Precision::Pconfusion()` to avoid missing possible intersections;
26664	*Summary:* Triangulating a very small polygon fails. `BrepMesh_FastDiscret::Parameters` now groups all all meshing parameters used to define and manipulate parameters of the algorithm. New parameter `adaptiveMin` from `BrepMesh_IncrementalMesh` provides adaptive computation of minimal 2D meshing precision.
26692	*Summary:* `BRepMesh` hangs on the attached shape because tessellation points are produced out of surface range. `BRepMesh_EdgeTessellator` avoids points out of face range using edge tolerance.
27119	*Summary:* Draw command `incmesh` hangs on the attached face. `BRepMesh_CircleTool::MakeCircle` now extends the circle radius by a small value to classify points forming the inscribed triangle as lying on it. `PConfusion` is not used as it can give a false positive result that hangs the algorithm.

Shape Healing

24658	*Summary*: `fixshape` changes the source shape. The context is used in method `ShapeFix_Wire::FixSelfIntersection()` to avoid modification of the original shape by shape healing. The check for sub-shape modification has been added in `XSAlgo_AlgoContainer::MergeTransferInfo()`.
26620	*Summary:* Shape healing unreasonably downgrades face tolerance. Methods `ShapeAnalysis_Edge::CheckSameParameter` and `ShapeFix_Edge::FixSameParameter` now consider only pcurve on a given face.

26642	*Summary:* `ShapeUpgrade_UnifySameDomain` introduces extremely high vertex tolerance. The algorithm computing safe shift value along 2D-line (to have the distance between two points less than tolerance) has been improved in class `IntCurve_IntConicConic`.
26656	*Summary:* `ShapeFix_Face` introduces extremely high vertex tolerance in the input shape. Methods `CopyVertex` have been added in `BrepTools_ReShape`. These methods are used by `ShapeFix_Wire` and `ShapeFix_Edge`.
26708	*Summary:* Fix shape did not fix the attached shape. Seam edge was not added to the attached periodic face. The crossing of a seam edge during collection of the wires is now taken into account in `ShapeFix_ComposeShell.cxx`.
27078	*Summary:* Exception in `ShapeFixIntersectionTool::UnionVertexes()`. Checks for presence of parametric curves have been added in method `ShapeFix_IntersectionTool::UnionVertexes`.

Samples

12042	*Summary:* Problem with standard Qt-based Import/Export sample application Front and right views in CSharp, mfc and qt samples have been swapped to have the same behavior as in Qt-based Import/Export application.
16472	*Summary:* Improve environment scripts for samples The environment setting scripts in OCCT samples now do not require defining `CASROOT` variable globally. It is possible to use local and relative paths. To launch Qt samples in the default environment execute OCCT/env.bat file without any arguments. Add compiler version, platform and mode (release or debug) as arguments to use samples in a specific environment. Correspondingly, CSharp and MFC samples can be launched using only the name of the sample as argument for the `run.bat` file, e.g. `run.bat Ocaf` or using more arguments to indicate a specific environment, e.g. `run.bat vc10 win64 Release Ocaf`.
24665	*Summary:* Sample for advanced function mechanism. New sample application `FuncDemo` illustrates usage of recently upgraded Function Mechanism of Open CASCADE Application Framework: ▪ construction of a graph of functions and iteration through the graph for single- and multi-threaded calculation modes; ▪ calculation of several simple models in single and multi-threaded modes.

26603 27291	*Summary:* Problem with maximization and normalization document windows in Qt samples with Qt 5.x. The problem with a failure to resize objects when the view area is maximized has been fixed in Qt samples.
26210	*Summary:* Sample Modeling: operation `Make revol` has text in window caption “Make a prism”. The caption text of `revol` operation has been corrected.
26260	*Summary:* Tcl scripts for MBB Gehause Rohteil and ANC101. Two new sample scripts `MBBGehauseRohteil.tcl` and `ANC101.tcl` have been added. These scripts generate shapes used by Computer Aided Manufacturing International (CAM-I) to compare modeling systems in 1979 and in 1983.
26741	*Summary:* Problem with building samples and demo. Scripts and project files have been updated to the proper environment.
26787	*Summary:* Do not declare redundant macros `WNT, LIN, LININTEL, WIN32, WIN64` within sample project files. Unused macros `WNT, LIN, LININTEL, WIN32` and `WIN64` have been removed from sample project files. Defines `_WIN32, __linux__` and `LININTEL` have been removed from *.pro files of Qt samples.
27086	*Summary:* `jniviewer` - avoid duplicating viewer redraws. Redundant viewer updates within `GLSurfaceView::RENDERMODE_CONTINUOUSLY` mode are now avoided.

Configuration

10031 24002	*Summary:* Overall code and build procedure refactoring The OCCT code has been upgraded to remove CDL and WOK: ▪ WOK-generated header files from `inc` folder and sources from `drv` have been moved to `src`. ▪ CDL files have been removed; however, the corresponding documentation comments have been inserted in the code. ▪ All packages have been converted to `nocdlpack`. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.

24786 26467 26559 26602 27055 27056	*Summary:* Move the functionality of WOK command `wgenproj` to OCCT tool `genproj`. The generation of build scripts and project files is now provided by the the tcl script adm/genproj.tcl (wrapped by batch scripts `genproj.bat` and `genproj.sh`) replacing WOK command `wgenproj` in earlier versions of OCCT. The tool for configuring `custom.bat` has also been copied; it is called by `genproj` if `custom.bat` does not exist yet. Environment variable `SHORTCUT_HEADERS` can be defined in `custom.bat` to put in `inc` folder the shortcuts to actual headers, instead of their copies.
24816 26919 27054	*Summary:* Tool for upgrade of OCCT and dependent code. New script `adm/upgrade.tcl` defines Tcl procedure `upgrade`, used for upgrading the code of OCCT and applications for changes introduced by OCCT 7.0. Batch script `upgrade.bat` is provided for convenience. The file `upgrade.dat` contains data (lists of classes) required for some upgrade steps. See the Upgrade Guide for usage recommendations.
25114 26475	*Summary:* CMake build tools for OCCT 7.0. CMake scripts are now completely included in OCCT sources so that CMake can be used to build OCCT directly.
26247	*Summary:* CMake – strip symbols within `Release` configuration when using `non-msvc`. Parameter `-s` has been added to `CMAKE_CXX_FLAGS_RELEASE` and `CMAKE_C_FLAGS_RELEASE` to optimize the size of binaries.
26388	*Summary:* Setting debug environment of the Visual Studio solution of OCCT generated by CMake. The following modifications have been introduced to properly set the debug environment of OCCT Visual Studio solution generated by CMake: ▪ Debugging environment has been added to `DRAWEXE vcxproj` ▪ Short-cut files for all OCCT headers are now collected in `<binary dir>/inc` during CMake configuration process. ▪ Tcl installation procedure installs all `.dlls` found in `tcl bin` folder (on Windows this approach takes into account `zlib` library that may be located in `tcl bin` folder). ▪ Each OCCT project includes only 3rd party paths and `<cmake binary dir>/inc` folder.
26389	*Summary:* The OCCT Visual Studio solution generated by CMake should have all build configurations. The following modifications are related to build configurations of OCCT Visual Studio solution generated by CMake: ▪ The OCCT build type now can be chosen in the generated projects. ▪ The search for debug library has been removed from 3rd-party search mechanism. ▪ `Draw.bat` now can be launched with three arguments defining compiler, bitness and build type.

26527	*Summary:* Some additional items in the link command of an OCCT product project have incorrect paths. The path for OCCT-product toolkit has been removed and paths of external linked libraries (`CSF_`) corrected.
26529	*Summary:* CMake offers rebuilding of platform dependent code unclear for a user. The following modifications have been introduced to improve rebuilding of platform dependent code by CMake: ▪ Descriptions of CMake variables have been updated; ▪ `TestCases` project and all related variables have been replaced by a custom script for building environment; ▪ `DrawAppliInit` is now copied from `occt` root to `CMake` binary directory if the file exists; ▪ Flex & Bison compiler flags have been added; ▪ `RelWithDebInfo` OCCT libraries have been moved to `libi` folder and the binaries to `bini` folder; ▪ `CMAKE_BUILD_TYPE` is released by default for a single configuration generator; ▪ The value of `CASDEB` in `env.bat` now equals by default to the build type of the last installed OCCT libraries.
26543	*Summary:* `genproj` script parses `CSF_VTK` incorrectly. The name of associative array in method `osutils:csfList` has been defined as `aCsfMap` (instead of `aLibsMap`).
26546	*Summary:* `genproj.tcl` – add support for VS2015 in project file generator. Generation of project for Visual Studio 2015 has been implemented.
26591	*Summary:* Header collecting in CMake configuration process does not consider `patch` directory. The `patch` directory is now properly taken into account by CMake configuration process. CMake variable descriptions have been updated and their width realigned.
26592 26652	*Summary:* Macros `OCCT_DEBUG` is not supported in configuration of 7.0.0 `dev` version with CMake. New variable `BUILD_WITH_DEBUG` enables extended messages of many OCCT algorithms, usually printed to cout, including messages on internal errors and special cases encountered, timing etc.
26600	*Summary:* CMake should disable auto-link for TBB. Implicit linkage with TBB has been disabled in the CMake configuration.
26615	*Summary:* `genconfdeps.tcl` – do not search for `FreeImagePlus` on non-Windows. The script `genconfdeps.tcl` now avoids searching for `FreeImagePlus` on non-Windows platforms.

26618	*Summary:* CMake should disable ability to use 3rdparty if there is no any included toolkit that can use it. The offers to use Freeimage, GL2PS and TBB are enabled dynamically. E.g. `USE_FREETYPE` variable is not offered if the toolkits using it (`TKOpenGl`, `TKService`, `TKV3d` and `TKViewerTest`) are not built.
26648	*Summary:* Set the default path to start `DRAWEXE` under Visual Studio debugger. The default path for starting executable from Visual Studio debugger has been set to `$CASROOT` in project files generated by `genproj`. The default path for starting `DRAWEXE` executable from Visual Studio debugger has been set to CMake binary directory in project files generated by CMake.
26763 26858	*Summary:* CMake configuration process does not allow setup paths for TK library if it is separated from TCL. TK has been separated from TCL in CMake configuration process and can be used without it. TCL/TK searching works correctly with installed ActiveTcl and latest versions of CMake.
26822	*Summary:* `OpenGl_Texture` – fix compilation issue on Android due to usage of undefined macros `GL_DEBUG_TYPE_ERROR`. OpenGl package has been revised to remove prefix `_ARB` from the enumeration `GL_DEBUG_`.
26823	*Summary:* Use EGL on another platform without GLX. The problems connected with OCCT usage on QNX platform have been fixed.
26830	*Summary:* `TKernel` should be linked with `pthread` and `rt` explicitly only on Linux. `CMakeLists` have been modified to avoid linking `pthread` and `rt` on Android and QNX.
26831	*Summary:* Define `HashCode()` for `pthread_t` on Android. Explicit cast of `Standard_ThreadId` to `Standard_Size` has been implemented in method `BOPCol_ContextFunctor::HashCode()`.
26835 26836	*Summary:* CMake – `CSF_OpenGlLibs` variable is not specified `CSF_OpenGlLibs` has been specified in `genproj.tcl` for Android system.
26854	*Summary:* Use `-Wextra` with GCC. The option `-Wextra` is now set for GCC compilation. This allows seeing additional warnings that can help to spot problematic places in the code.
26861	*Summary:* CMake – enable `-Wall` warnings when using Clang. `-Wextra` warning option is now used for Clang compiler.

26862	*Summary:* Avoid usage of 3rd-party headers within `OpenGl_View` and `D3DHost_View`. The following modifications have been implemented to avoid 3rd party code inclusion (FreeType and D3D9) within `OpenGl_View` and `D3DHost_View`, which causes problems during compilation: ▪ The structure `Font_FTFont::Rect` has been moved in the dedicated header `Font_FTFont`. ▪ Forward declarations are used in `D3DHost_View`, `OpenGl_Font` and `Font_TextFormatter`. ▪ Method `D3Dhost_View::Redraw()` now assigns `myFBO` before rendering. ▪ The problem with always zero viewport has been fixed in method `D3DHost_FrameBuffer::Init()`.
26865	Summary: CMake – provide consistency between `FILES` and the actual content of `inc` and `src` folders. The following script modifications have been introduced to provide consistency between `FILES` and the actual content of `inc` and `src` folders: ▪ Headers from `inc` folder are checked for use in further building process; ▪ `FILES` file is parsed to collect header files for `inc` folder; ▪ The headers with name not containing their package names are not removed during `inc` folder cleaning. A warning is made when a file in folder is not listed in the corresponding `FILES` file.
26867	*Summary:* Update any comparison including `__cplusplus` macro to remove it. The macro `_QNX_` has been implemented in methods `ElCLib::HyperbolaParameter()`, `Standard_Real::ACosh` and `Standard_Real::ATanh`.
26868 26875 26903	*Summary:* CMake – 3rdparty search algorithms should have the same logic. The algorithms searching for freeimage, freetype, gl2ps, tbb and vtk have been revised to use similar logic. Additionally: ▪ `3rdparty_dir` is not specified by default; ▪ `3rdparty_<name>_dir` has priority over `3rdparty_dir`; ▪ The parsing of package `FILES` is ended if all files from it have been processed; ▪ X11 package is searched for whenever it is used on OS X.
26873	*Summary:* Tcl 8.6.2 crashes. Tcl/Tk has been upgraded from version 8.6.2 to 8.6.4 to resolve problems with wrong use of GDI resources.
26878	*Summary:* CMake - encode properly version of OCCT in Windows binaries. Generated resource files `*.rc` have been added to each toolkit project for msvc. They contain product version, copyright, and other relevant data.

26880	*Summary:* CMake – platform dependent code is not generated. CMake lists have been updated for each package. CMake now considers `CSF_` variables from `EXTERNLIB` file for each toolkit. `CSF_` variables are defined in `occt_csf` for each OS. Redundant CSF have been removed.
26902	*Summary:* CMake – tests is not installed after definition `INSTALL_OCCT_TEST_CASES` option. The variables responsible for installation of tests and samples have been corrected in CMake scripts.
26911	*Summary:* CMake – strips symbol information from the binary in release configuration if the compiler is a variant of GCC. The strip flag is now added when the compiler is a variant of GCC.
26916 27044 27062	*Summary:* Configure variables to customize paths for OCCT executables, libraries and resources. It has become possible to customize the installation path separately for executables, libraries and resources. `OCCT_RESOURCE_PATH` environment variable is used in paths.
26939	*Summary:* `NCollection_UBTreeFiller` - do not use `_REENTRANT` in a header file. Random number generator `std::mt19937` is used instead of `rand()` in `NCollection_UBTreeFiller`.
26941	*Summary:* Building on Windows with VC14 - debug info is generated for Release build. Generation of debug info is explicitly disabled on link step in templates of Visual Studio 10+ project files used by genproj, to avoid relying on default settings of Visual Studio.
26993 27025 27040	*Summary:* CMake - use the abstraction level of VTK instead of `CSF_VTK` for their libraries. `CSF_VTK` has been removed from CMake meta-projects (this variable remains in `EXTERNLIB` because it is used by `genproj.tcl`). `USE_VTK` is shown if toolkits using VTK are involved in the solution. `VTK_DIR` has been removed from advanced variables.
27041	*Summary:* CMake - CLang 3.6.2 fails to link DRAWEXE on Ubuntu 15.10. "stdc++" has been added to `CSF_ThreadLibs` in CMake script to properly build `DRAWEXE`. In addition, TKVCAF has been added in CMake script for OCAF MFC sample, and Yacc and Lex files have been added in `src/StepFile/FILES` to make it consistent with actual contents of the package.
27061	*Summary:* Compilation failure due to BOM in `OpenGl/OpenGl_BVHClipPrimitiveTrsfPersSet.cxx` The byte order mark (BOM) has been removed in `src/OpenGl/OpenGl_BVHClipPrimitiveTrsfPersSet.cxx` to allow compilation on some gcc compilers.

27062	*Summary:* Misspelling of resources install path in `CMakeLists.txt`. An error in resources install path has been fixed.
27095	*Summary:* CMake – configuration process finds `tcl` folder with incorrect bitness for 32-bit case A new pattern for seeking 3rdparty products has been added.
27121	*Summary:* CMake – build process does not use patched header files. The script collecting all headers to `<binary dir>/inc` folder now takes into account patched header files.
27152	*Summary:* Undefined symbols in library `TKernel.so` using clang compiler. The following dependencies and options have been added: ▪ `dl` dependency for TKernel only; ▪ `lm` dependency for all toolkits.
27197	*Summary:* Fix compilation issues when using `mingw`. The problems specific for OCCT compilation using `mingw` have been fixed.

Coding

22928 22972	*Summary:* Eliminate macro definitions that have compiler-provided analogs (`WNT`, etc.). Macro definitions required for successful compilation on different platforms have been replaced with appropriate compiler-provided macro definitions: ▪ `WNT` with macros `_WIN32` and `_MSC_VER` for platform and compiler detection accordingly. ▪ `LIN` with macro `__linux__`. ▪ `DEB` with macro `OCCT_DEBUG`.
24567 24875 25078	*Summary:* Coding rules – eliminate GCC warnings. OCCT code has been revised to remove the following warnings: ▪ `-Wignored-qualifiers` - redundant `const` qualifiers of return types of functions returning values; ▪ `-Wstrict-aliasing` in method `OpenGl_TriangleSet::Box();` ▪ `-Wclobbered` - suppressed by #pragma in `Standard_ErrorHandler.hxx` when `OCC_CONVERT_SIGNALS` or `NO_CXX_EXCEPTIONS` are used.
24750 26850	*Summary:* Replace instantiations of `TCollection` generic classes by `NCollection` templates. The instantiations of generic collection classes from `TCollection` have been replaced by equivalent instantiations of `NCollection` template classes. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
24859	*Summary:* Replace `SortTools` by STL equivalents. Package `SortTools` and its derived classes have been removed and replaced throughout OCCT by C++ STL sorting algorithms (i.e. `std::sort`). Comparator objects have been implemented as local classes. See also the Upgrade Guide for the information about possible impact of this change on the existing applications and relevant porting recommendations.
24870	*Summary:* Provide OCCT RTTI test cases. Test procedures for checking the performance and functionality of OCCT handles and RTTI have been added.
24895	*Summary:* `PLUGINFACTORY` has C-linkage specified, but returns a user-defined type `Handle_Standard_Transient`, which is incompatible with C. The definition of `PLUGINFACTORY` function returns `Standard_Transient*` instead of `Handle(Standard_Transient)`. The default implementation of `PLUGINFACTORY()` instantiated by macro `PLUGIN()` has been corrected accordingly. Methods `Factory()` in persistence packages now return `const &` to handle.
24967 26595	*Summary:* Documentation – lost some comments in OCCT code after cdl elimination. Comments in instances of generic classes previously lost during generation of .hxx files have been recovered.

25076 26990	*Summary:* Hidden overloaded virtual functions. Implementation of virtual functions has become more consistent: ▪ Missing implementation of virtual method Closed() has been added in classes inheriting `Intf_Polygon2d`; ▪ Empty implementation of virtual method `Read()` accepting stream has been moved from `PCDM_RetrievalDriver` to `StdLDrivers_DocumentRetrievalDriver`; ▪ Method `BrepFill::Delete()` has been renamed to `DeleteProfile()` to avoid confusion with method `Delete()` inherited from `MMgt_Tshared`; ▪ Virtual method `AIS_Dimenaion::ComputePlane()` has been removed from base class; each dimension defines and uses its own method with the same name (but different arguments); ▪ Inherited virtual method `Dump()` with a single argument in class `XCAFDoc_ShapeTool` is now defined as short-cut to own method `Dump()`, also calling parent's one; ▪ Inherited virtual method `BoundingBox(void)` is made visible in `AIS_Shape`; ▪ Inherited virtual method `Box(void)` is made visible in classes inheriting `BVH_PrimitiveSet.` ▪ Signature of methods `xsputn()` and `overflow()` from class `LDOM_SBuffer` now corresponds to the signature of overridden virtual methods of `std::streambuf`.
25454 26042	*Summary:* OCCT does not work with the latest Xcode. Use of `NULL` references has been eliminated in classes `Plib`, `BsplCLib` and `BsplSLib`.
25571	*Summary:* Avoid base Classes without virtual Destructors. Destructors of collection classes provided by `NCollection` and `math_Function` are made virtual. This allows safe destruction by pointer to base class. Destructors of classes `HatchGen_IntersectionPoint`, `IntCurveSurface_Intersection`, `Intf_Interference` and `IntRes2d_Intersection` are made protected to avoid destruction by pointer to the corresponding base class.
25617	*Summary:* Avoid classes with a copy constructor and the default destructor or assignment operator. Useless user-defined copy constructors and assignment operators have been removed from classes `BOPCol_NCVector`, `NCollection_Mat4` and `NCollection_Vec*`. A user-defined assignment operator matching the copy constructor has been added in `NCollection_StdAllocator`. The class `VrmlData_DataMapOfShapeAppearance` has been redefined as a simple `typedef` to `NCollection_DataMap<>`.
25618	*Summary:* Avoid classes with a copy constructor and the default destructor or assignment operator. A user-defined assignment operator has been removed from classes `IntPolyh_StartPoint` and `Quantity_Color`, because the default assignment is enough.

26178	*Summary:* Coding rules – eliminate `-Wtautological-pointer-compare` Clang warnings in `Standard_ErrorHandler`. Tautological comparisons have been removed in `Standard_ErrorHandler` and Standard_ErrorHandlerCallback.
26179	*Summary:* Coding rules – eliminate `–Wdeprecated`-declarations Clang warnings on `tmpnam()` usage. New method `OSD_File::Capture()` has been implemented for standard output redirection. Temporary files are now created using method `OSD_Directory::BuildTemporary` in folder `/tmp` on Linux or using `TEMP` environment variable on Windows.
26207 26668	*Summary:* Eliminate compile warnings obtained by building OCCT with vc14: conversion requires a narrowing conversion. The following improvements have been introduced to eliminate compiler warnings: ▪ `Ivtk_IdType` has been defined via `vtkIdType` to eliminate warnings "conversion requires a narrowing conversion". ▪ It is checked if OCCT and VTK use the same bitness. ▪ `HashCode()` function has been added in `Standard_Integer.hxx` to handle 64-bit integers.
26506	*Summary:* Change class `BRepLib_CheckCurveOnSurface`. The geometric part of class `BRepLib_CheckCurveOnSurface` and try/catch processing have been moved to `GeomLib_CheckCurveOnSurface`.
26581 26583 26584 26669	*Summary:* Eliminate compile warnings obtained by building OCCT with vc14. The OCCT code has been revised to eliminate the following vc14 compiler warnings: ▪ type cast conversion; ▪ declaration of local variable hides function parameter; ▪ declaration of variable hides class member; ▪ type cast: conversion from `BOOL` to `WNDPROC` of greater size.
26585	*Summary:* Eliminate compile warnings obtained by building OCCT with vc14: 'type cast' pointer truncation and 'type cast' truncation. The following modifications have been introduced to eliminate vc14 compiler warnings: ▪ Class `OSD_EnvironmentIterator` has been removed; ▪ `Draw_ProgressIndicator` has been corrected to properly pass address via Tcl; ▪ `OSD_File` has been refactored to avoid senseless encoding / decoding of results; ▪ Methods `OSD_FileNode::UserId()`, `OSD_FileNode::GroupId()` and `OSD_Process::UserId()` have been removed, as they cannot be cross-platform; ▪ `OSD_Thread` now uses WinAPI conversion functions to avoid warnings; ▪ Recursion counter is passed in `OSD_WNT` via function argument instead of TLS; ▪ The class `TDF_LabelMapHasher` has been revised to use the correct hasher function for an address.

26690	*Summary:* Compilation problem on iMac Monobloc. The declaration `parseOnOff()` has been implemented in `ViewerTest::ParseOnOff()` as a static method.
26780	*Summary:* Coding rules – eliminate warnings on Linux and Mac. A few warnings found by GCC, Clang, and VC++ 14 have been fixed by use of `ifdef/ifndef` or removal of unused code.
26781	*Summary:* Coding rules – eliminate GCC warning `–Wunused-result`. The results of fgets() and system() functions are now checked in methods `FSD_BinaryFile::ReadChar()`, `IFSelect_SessionFile::ReadFile()`, `IFSelect_SessionPilot::ReadScript()`, `OSD_File::Print()`, `OSD_Process::Spawn()`, `RWStl::ReadAscii()` and `iges_lire()`.
26783	*Summary:* Coding rules – eliminate GCC warning `–Wunused-but-set-parameter`. Assigned parameters are now passed by reference in methods `GetTol()` and `GetPar()` from class `TestTopOpeDraw_Displayer`. Unused code has been removed.
26784	*Summary:* Coding rules – eliminate GCC warning `–Wunused-parameter`. In `OSD_Thread` class, `pthread_join()` has been replaced by `pthread_timedjoin_np()` when available (`glibc` extension). Unused parameter warning has been eliminated in `OSD_Signal`, `NCollection_WinHeapAllocator`, `OpenGl_Text`, `OpenGl_View`, `V3d_View` and `ViewerTest`.
26785	*Summary:* Coding rules – eliminate GCC warning `–Wempty-body` in `LDOM_DeclareSequence.hxx`. A GCC warning been fixed in class `LDOM_DeclareSequence`.
26788	*Summary:* Compiler warnings when `OCCT_DEBUG` is enabled. The code has been revised to eliminate warnings occurring when `OCCT_DEBUG` is enabled. Additionally, `PPC` variables have been renamed to avoid conflicts on PowerPC.
26805	*Summary:* `Graphic3d_GraphicDriver` – drop outdated unsupported methods for debugging. The following unsupported methods have been removed from `Graphic3d_GraphicDriver` class: `PrintBoolean()`, `PrintCLight()`, `PrintCStructure()`, `PrintCView()`, `PrintFunction()`, `PrintInteger()`, `PrintIResult()`, `PrintShortReal()`, `PrintMatrix()`, `PrintString()`, `SetTrace()` and `Trace()`.
26811	*Summary:* Coding rules – drop unused Draw Harness command `deboucle`. Unused Draw command `deboucle` has been removed from `BrepTest_FeatureCommands` class.

26843	*Summary:* New warning during compilation OCCT on OS X. Unused (and not implemented) methods `IsDivisible()`, `GetExponent()`, `GetMantissa()` and `AvailableMemory()` have been removed from `OSD` package.
26852	*Summary:* Coding – compiler warnings issued by GCC 5.2.1. The following modifications have been introduced to eliminate GCC compiler warnings: ▪ Copying of local list is avoided in class `BrepAlgo_DSAccess;` ▪ "Possibly used uninitialized" variables are initialized by zeros in classes `IntPatch` and `IntTools`: ▪ Unused argument `theContext` has been removed from method `OSD_signal::SegvHandler`; ▪ Missing initializers have been added in OpenGL structures; ▪ Function signature has been corrected in class `STEPConstruct_GDTProperty` to pass output parameters by reference.
26872	*Summary:* Pointless instantiations of local variables in `BinTools.` In `BinTools` package, the instantiations of local variable `stringstream`, which is used in error handling, have been moved from the upper function scope to where they are actually needed.
26912 27080 27091 27106	*Summary:* Clang compiler warnings. OCCT code has been revised to get rid of the following warnings: ▪ `-Winconsistent-missing-override` - about missing override specifier in all declarations of virtual methods in descendant classes; ▪ `unused parameter on OS X` ▪ `-Wall` warnings – missing overrides and undefined internal linkages
27050	*Summary:* License is not activated on OS X. On OS X host id now gets as a hash for char array returned by `gethostuuid()`.
27057	*Summary:* Wrong license statements in some files. Copyright statements in OCCT code have been updated.
27067	*Summary:* Avoid use of virtual methods for implementation of destructors in legacy classes. Redundant methods `Delete()` and `Destroy()`, created in CDL as a hack to define destructor for the class, have been removed; their definitions converted to the definition of destructors. In some places methods `Destroy()` are still preserved (bug made non-virtual) because they are called explicitly.
27068	*Summary:* Eliminate VC++ 14 compiler warnings in MFC samples. The code of MFC samples has been revised to avoid compiler warnings.

27097 27105	*Summary:* Make code ISO-compliant `[-Wpedantic]` fixes. The code has been revised for consistency with CLang and GCC `-pedantic` ISO compliance option: ▪ extra semicolons after closing braces of namespaces and macros removed; ▪ function prototypes corrected; ▪ unnecessary declarations of system functions removed.
27113 27118	*Summary:* Add macros `Standard_DEPRECATED` for marking deprecated functionality. Macro `Standard_DEPRECATED` can be used in declarations to mark a method as deprecated and generate a compiler warning when it is used. If `OCCT_NO_DEPRECATED` is defined, `Standard_DEPRECATED` is disabled (defined empty).
27266	Summary: `TKOpenGl` - drop unused files Unused files `OpenGl_telem_util.hxx` and `OpenGl_tgl_funcs.hxx` have been removed.

Documentation

23645	*Summary:* User's Guide about Modeling Algorithms contains a wrong statement about `GeomFill_ConstrainedFilling`. Wrong statement about `GeomFill_ConstrainedFilling` has been removed.
24514	*Summary:* Unclear guidelines to report issues in Mantis. Contribution workflow document is revised and extended to eliminate inconsistencies and describe the process in more details: ▪ Meaning of issue fields in Mantis (Category, Severity, Profile, Project and Target Version, etc.) is described in more details; ▪ Additional rules are defined and examples given for defining Summary and Description, and writing commit messages; ▪ Requirements for testing, creation of a test case, update of user documentation, etc. when resolving an issue, are described; ▪ Additional elements of the workflow (patch submission, rebasing branches, use of feedback status, issue relationships) are described.
26187	*Summary:* Implement m-dashes in the documentation. Documentation has been revised to implement m-dashes and remove Unicode dashes.
26212	*Summary:* There are some problems with location of text and images on pages Images have been redesigned or resized to fit on page in generated PDF documents.
26231	*Summary:* Update description of selection algorithm in the user's guide. The description of selection algorithm in visualization user's guide has been updated.

26478	*Summary:* Documentation on OCAF / Topological naming. A section about topological naming mechanism has been added.
26488	*Summary:* The class `ShapeUpgrade_UnifySameDomain` is not documented. The class `ShapeUpgrade_UnifySameDomain` has been documented in details.
26537	*Summary:* It is not possible to generate reference documentation in new structure of OCCT. OCCT products reference documentation has been fixed by changing the path to search for required headers from `<occt>/inc` to `<occt>/src/<each package>`.
26744	*Summary:* Deviation angle default value as stated in AIS_InteractiveContext.hxx is wrong. The default value of deviation angle has been corrected in class documentation.
26799	*Summary:* `RowLength` and `ColLength` return a wrong value. Comments to methods `NCollection_Array::RowLength()` and `ColLength()` have been improved to avoid possible misinterpretation.
26804	*Summary:* The level of most VIS Viewer commands is incorrect. The structure of headings in Draw User's guide has been corrected.
26807	*Summary:* Describe dropping of Visual3d and UserDraw within porting notes. A new developer guide Upgrade from older OCCT versions has been added in the documentation to provide guidelines for update from earlier OCCT versions to 7.0.0.
26869	*Summary:* Simplify the developer guide "Building with CMake and ADT on Android". The instruction for OCCT building on Android has been updated.
26926	*Summary:* Small mistake in the exceptions part of Foundation Classes User's Guide. The incorrect statement has been fixed.
26964	*Summary:* Merge OCAF white-papers into OCAF user's guide. OCAF white-papers have been merged into OCAF user's guide to provide a single source of information on the subject.
26970	*Summary:* Update MSVC visualizers to support 7.0 handles. The presentation of variables of type `opencascade::handle`, i.e. `Handle(Class_Type)`, and several other classes in the Visual Studio debugger has been improved.
26992	*Summary:* Use Doxygen tag @ref for cross-references in documentation. Doxygen tag @ref has been implemented uniformly for cross-references within the documentation.

27023 27072	*Summary:* Highlight links in PDF by color. Cross-references in the text of generated PDF documents are now highlighted by color and followed by the reference page number.
27088	*Summary:* Documentation of add method of `GeomConvert_CompCurveToBSplineCurve` lacks speed hint. Description of `WithRatio` parameter has been added in `GeomConvert_CompCurveToBSplineCurve.hxx`.
27120	*Summary:* Documentation for check commands. Documentation for commands `checkprops`, `checkdump` and `checklength` has been added.
27193	*Summary:* Describe building OCCT with genproj tool. The description of OCCT building with WOK and automake scripts has been removed from the documentation and replaced by the instructions for the use of `genproj` utility. New script `genconf` allows starting the configuration GUI explicitly.

WOK

22827	*Summary:* Make non-CPP source files (CDLs, headers) to appear in MS VS project files. Generation of OCCT overview documentation has been added in Visual Studio projects generated by CMake, as `Overview` project. `BUILD_OCCT_OVERVIEW` variable provides generation of OCCT overview documentation in html format. Header files (*.h, *.hxx, *.lxx, *.gxx) have been included in Visual Studio projects.

Release

26242	*Summary:* OCCT Install Wizard must install VC2010 redistributable. OCCT Install Wizard now installs VC2010 redistributable if the binaries are installed.
27053 27138	*Summary:* Compilation fails with "fatal error RC1103: invalid option, `/fp:precise`". CMake script has been corrected to add compiler options (besides macro definitions) to `CMAKE_CXX_FLAGS` and `CMAKE_C_FLAGS` variables directly, instead of using `add_definitions()` command. This eliminates the build failure by NMake due to incorrect options passed to RC compiler.

Added-value components

ACIS-SAT Import / Export

26826	*Summary:* Exception on export to ACIS. Exception caused by accessing an empty `Sequence` has been fixed in method `AcisData_CasCadeToAcis::Wire`.
26909	*Summary:* The file exported from AutoCAD does not show any colour data when importing it using XDE in OCCT. `SATCAFControl::DecodeColor` now can decode color attribute `AcisAttr_AttribTruecolAdesk`.
27248	*Summary:* Reading some DXF files gives incorrect results. Support of new types `cl_loft_spl_sur` and `sweep_sur` has been introduced in ACIS reader.

Parasolid Import

26982	*Summary:* Exception on reading of unknown colour representation. Method `XtCAFControl_Reader::ReadColors` can process and reject unknown colors.
26983	*Summary:* Incorrect reading of long pointers. Pointer values are now stored as `ExtCharacter` (instead of `Integer`).

DXF Import / Export

26769	*Summary:* Chinese letters are not translated. DXF Sample now can read the font from file, instead of using a fixed font. Chinese symbols can be translated for `MText` entities.
26856	*Summary:* Polyline loses one of its coordinates. Polyline elevation has been added for 2d polylines in class `DxfData_MakePolyline`.
27204	*Summary:* Some types of DXF surfaces are not supported. DXF reader now can process all `SURFACE` types.

Best Fit

26849	*Summary:* Problem with incorrect mapping of points with non-null offset on two-sided models. The algorithm of gradient calculation has been fixed in method `BestFitAlgo_Function::gradient()`.

Surfaces from Scattered Points

27084	*Summary:* Colors and Color scale are not displayed while morphing a surface. SSP sample now enables color levels for the displayed shape and shows Color scale when Surface morphing is activated.

Mesh Framework

26680	*Summary:* Changed behavior of mesh visualization and selection in OMF sample. `MeshVS_Mesh` now can handle the global selection mode.
27163	*Summary:* Correct point projection to mesh elements. `OMFAlgo::ProjectPointOnMesh` correctly projects any given point on the mesh elements. Draw command `MFpoint2mesh` has been created for the algorithm.
27173 27220	*Summary:* Optimize binary STL reader. The buffered file reading has been enabled in binary STL reader. Consistent support of UNICODE paths has been introduced.
27205	*Summary:* Add import and export of OBJ format to Qt OMF sample. Class `OMFTools_OBJFile` has been extended with functionality of translating data to and from `OMFDS_Mesh` and saving it to `.obj` file. Import and export to OBJ format have been added to OMF sample.
27219	*Summary:* Optimize ASCII STL reader ASCII STL reader has been optimized to provide a more efficient parsing of rows read from file. Consistent support of UNICODE paths has been introduced.
27249 27289	*Summary:* Adding timer for all operations. The timer has been implemented for all operations in Qt version of Mesh Framework sample to show how much time they take.

27278	*Summary:* Add Section operation to the Qt sample application. "Mesh Section" operation has been implemented in Qt version of Mesh Framework sample to allow computing sections between two meshes.
27308	*Summary:* Optimize OBJ reader and writer. OMF OBJ writer now provides more efficient access to mesh element nodes. OMF OBJ reader now provides more efficient parsing of lines read from file.

Advanced Samples & Tools

26391	*Summary:* Add WPF sample to the Advanced C# Wrapper tool. New `ImportExportWPF` sample represents OCCT 3D viewer integration into a single document WPF application using `D3DHost.D3DHost_ImageView` component.
26753	*Summary:* Unify code reporting license information in product samples. The information on available licenses and the status of license file is now shown in About dialog of samples.
26761	*Summary:* Not all static parameters are accessible in the XDE sample GUI. XDE sample now allows modifying some static parameters in its GUI.
26813	*Summary:* Problem of output in console during manipulation with mouse in the ImportExport sample of the Advanced Java Wrapper tool. The console messaging during manipulation with mouse has been eliminated in Java Import\Export sample.
26888	*Summary:* Crash in the `ImportExport` sample of the Advanced Java Wrapper tool on shape deleting. The behavior of 3D viewer and toolbar buttons after shape deleting has been corrected.
27096	*Summary:* Advanced C# Wrapper - add descendants of `V3d_Light`. Descendants of `V3d_Light`: `V3d_AmbientLight`, `V3d_PositionLight` and `V3d_PositionalLight` have been added to wrapper classes.

Open CASCADE Technology & Products

Upgrade to OCCT 7.0.0

Since version 7.0.0., known issues encountered during porting of OCCT and some derived applications as well as the approaches that have helped to resolve them are available in the online Upgrade developer's guide (http://dev.opencascade.org/doc/overview/html/occt_dev_guides__upgrade.html).

Supported Platforms and Pre-requisites

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

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

Linux Operating System	Mandriva 2010, CentOS 5.5, CentOS 6.3, Fedora 18, Ubuntu-1304, Debian 6.0, Debian 7.0
Windows Operating System	MS Windows 10 / 8 / 7 SP1 / Vista SP2 / XP SP3
Mac OS X Operating System	Mac OS X 10.10 Yosemite / 10.9 Mavericks
Android Operating System	Android 4.0.3 and above
iOS Operating System	iOS 7
Minimum memory	512 MB, 1 GB recommended
Free disk space (complete installation)	650 MB of disk space, or 1,4 GB if installed with reference documentation
Graphic library	OpenGL 3.3+, OpenGL ES 2.0+
C++ *For Linux:* *For Windows:* *For Mac OS X:*	 GNU gcc 4.3+ LLVM Clang 3+ Microsoft Visual Studio 2010 SP1 Microsoft Visual Studio 2012 Update 4 Microsoft Visual Studio 2013 Update 2 Microsoft Visual Studio 2014 Intel C++ Composer XE 2013 SP1 GCC 4.3+ (Mingw-w64) XCode 6 or newer
TCL (for testing tools) *For Linux:* *For Windows:* *For OS X:*	 Tcltk 8.6.3+ http://www.tcl.tk/software/tcltk/8.6.html Tcltk 8.6.3+ http://www.tcl.tk/software/tcltk/8.6.html or ActiveTcl 8.6 http://www.activestate.com/activetcl/downloads Built-in Tcl/Tk 8.6+
Qt (for demonstration tools)	Qt 4.8.6 http://www.qt.io/download/
FreeType (OCCT Text rendering)	FreeType 2.4.11-2.5.5 http://sourceforge.net/projects/freetype/files/
FreeImage (Support of common graphic formats)	FreeImage 3.17.0 http://sourceforge.net/projects/freeimage/files/
gl2ps (Export of OCCT viewer contents to vector graphic file)	gl2ps-1.3.8 http://geuz.org/gl2ps/
TBB (optional tool for multithreaded algorithms)	TBB 4.x or 5.x http://www.threadingbuildingblocks.org/
Doxygen (optional for building documentation)	Doxygen 1.8.5+ http://www.stack.nl/~dimitri/doxygen/download.html

