



Open CASCADE Technology and Products ver. 6.7.0

Release Notes

Overview

Open CASCADE Technology and Products version 6.7.0 is a minor release, which includes over **350** new features, improvements and bug fixes over the previous release 6.6.0.

Version 6.7.0 is binary incompatible with the previous versions of Open CASCADE Technology and Products, so applications linked against a previous version must be recompiled to run with this Version 6.7.0.

Highlights

- ➔ Dynamic clipping of shaded objects by arbitrary planes, with cross-section visualized by capping algorithm;
- ➔ Optimized and enhanced presentation of point markers using point sprites and VBO;
- ➔ Enhanced presentation of dimensions;
- ➔ Support of GLSL programs;
- ➔ Ray tracing mode of display of shaded objects in OCCT viewer providing high-quality image (including sharp shadows, correct transparency, reflections);
- ➔ Functionality to create BRep shape representing a text string (with specified font);
- ➔ Multiple corrections in Voxel package;
- ➔ New (more robust) 2D fillet algorithm;
- ➔ New documentation system: Doxygen is used to generate Overview and User Guides, their sources are included in OCCT sources;
- ➔ Considerably improved code stability, and clean-up of compiler warnings;
- ➔ Algorithm of result assembling in Boolean Operations is made parallel;
- ➔ Optimization: use of SSE2 instructions, refactored code of b-spline evaluation, revised XML persistence of binary data;
- ➔ Refactored CSharp sample, now including WPF front-end;
- ➔ Official support of Intel compiler, Visual Studio 2013, and Mac OS X 10.9 Mavericks



Table of Contents

New features	3
<i>Ray tracing as alternative rendering method for OCCT visualization component</i>	3
<i>Improved presentation of markers</i>	5
<i>Improved presentation of dimensions</i>	6
<i>Integration of shaders into OCCT visualization component</i>	8
<i>Object-level clipping planes with optional capping</i>	9
<i>Reflection line</i>	10
<i>Fillet algorithms based on Newton method and analytical equations</i>	10
<i>DXF Import Interface</i>	10
<i>Pipe construction from a swept shell with varying section width bounded by auxiliary spine</i>	11
<i>Documentation</i>	11
Modifications	12
<i>Foundation Classes</i>	12
<i>Application Framework</i>	14
<i>Modeling Data</i>	15
<i>Modeling Algorithms</i>	16
<i>Visualization</i>	27
<i>Data Exchange</i>	37
<i>Draw</i>	38
<i>Mesh</i>	40
<i>Shape Healing</i>	41
<i>WOK</i>	42
<i>Build</i>	43
<i>Samples</i>	47
<i>Release</i>	48
<i>Documentation</i>	48
<i>Products</i>	49
<i>Express Mesh</i>	49
<i>Advanced Samples and Tools</i>	50
<i>DXF Import-Export</i>	50
<i>ACIS-SAT Import-Export</i>	50
<i>PARASOLID-XT Import</i>	51
<i>Products Building</i>	51
Porting to version 6.7.0	52
<i>Object-level clipping and capping algorithm.</i>	52
<i>Redesign of markers presentation</i>	52
<i>Default views are not created automatically</i>	52
<i>Improved dimensions implementation</i>	53
<i>NCollection_Set replaced by List collection</i>	53
Supported Platforms and Pre-requisites	54

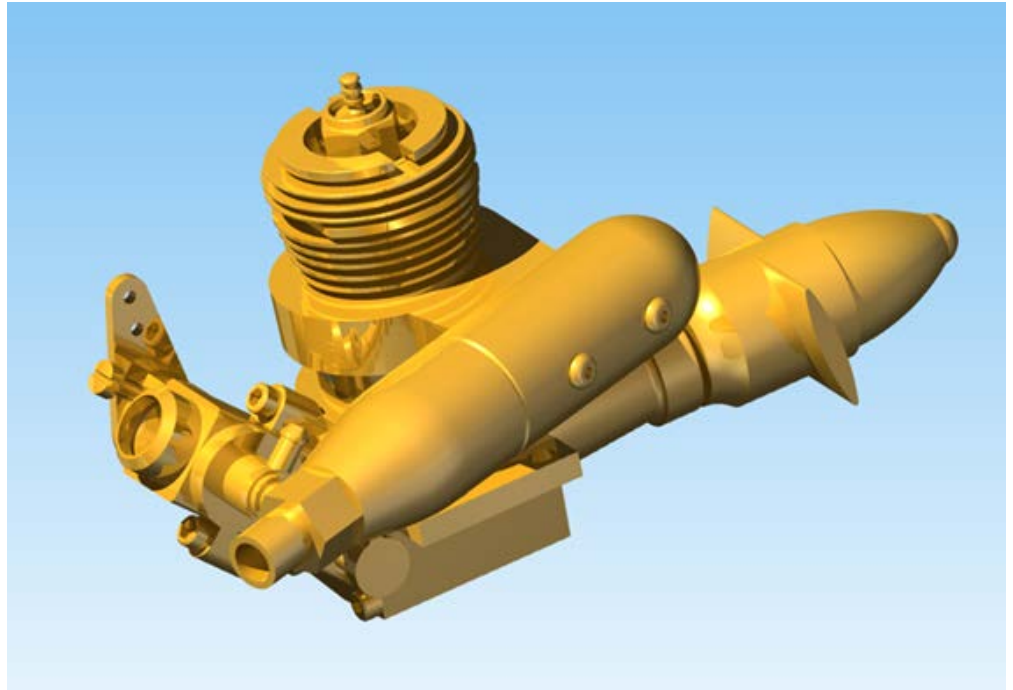
New features

Ray tracing as alternative rendering method for OCCT visualization component

The most breaking change in OCCT visualization introduced in OCCT 6.7.0 consists in simplified implementation of ray tracing algorithm for regular OCCT visualization primitives. This algorithm is implemented using OpenCL. As a part of TKOpenGL library, it is fully integrated with standard OCCT visualization logic. See [issue 24130](#) in Visualization.

From now on, ray tracing can be used as alternative rendering method instead of OpenGL rasterization.

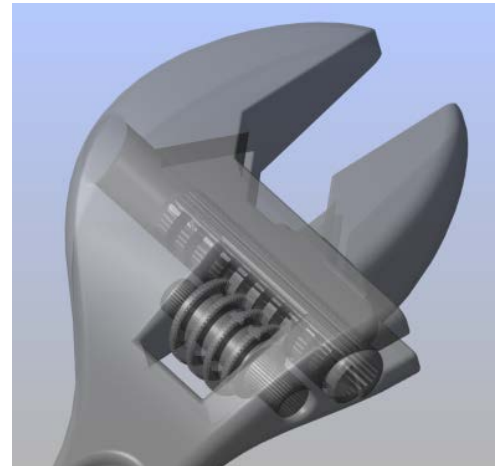
Compared to third-party ray tracing libraries, the built-in implementation provides real-time rendering performance, which is close to or even better than OpenGL rasterization performance on heavy models.



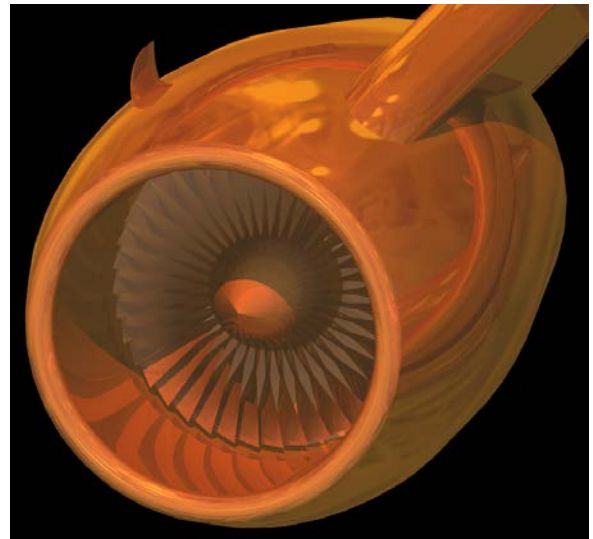
The simplified, yet powerful, ray tracer implementation has the following features:

- High-quality shading by interpolating normal across the triangle pixels and using advanced illumination model (Phong shading);
- Depth-correct order-independent transparency for any number of overlapped objects;
- Sharp shadows from isotropic point and directional lights without any approximation;
- Specular reflections and environment mapping for high-quality rendering of metallic surfaces;
- Low-cost adaptive anti-aliasing, which improves image quality by removing jagged edges from the image;
- Support of GPUs by major vendors (NVIDIA and AMD/ATI) owing to cross-platform OpenCL framework;
- Successful performance not only on top and mid-range GPUs, but also on low-end GPUs that support OpenCL.





Once OCCT has been built with OpenCL support, using ray tracing is really simple: it is enough to call `V3d_View::SetRaytracingMode()` to switch the 3D view to ray tracing rendering. Refer to `V3d_View` class documentation and check `vraytrace` and `vsetraytracemode` DRAW Test harness commands for details how to adjust ray tracing options.



The first version of ray tracing algorithm has some limitations to be addressed in the future OCCT releases:

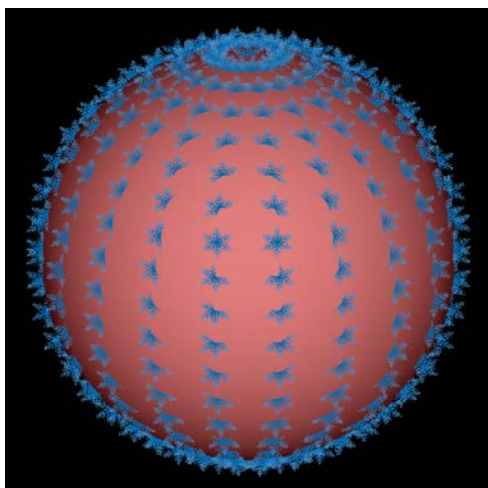
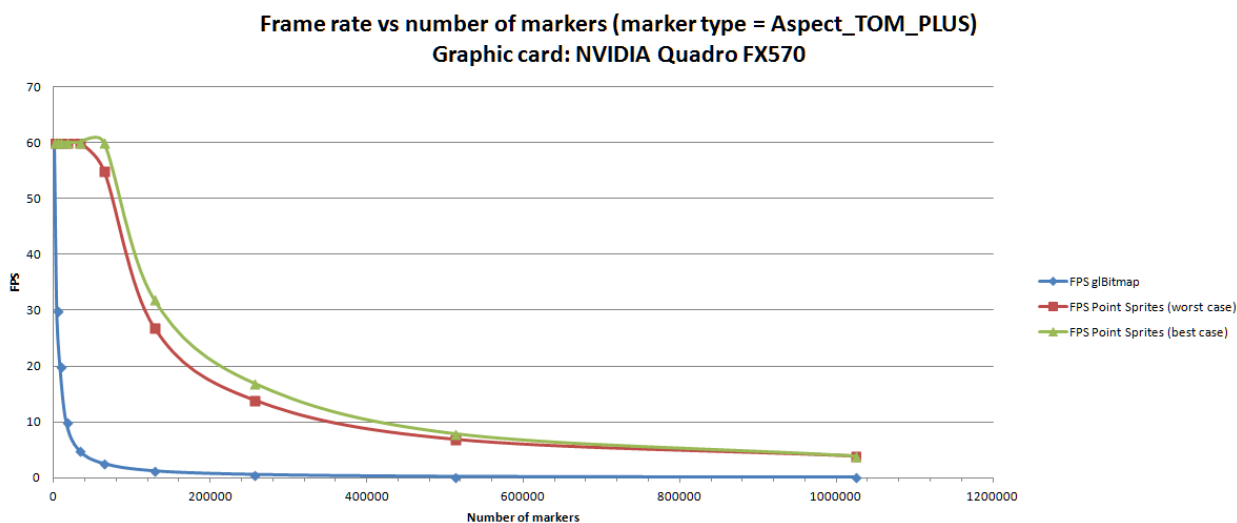
- Textures are not imported from OCCT visualization data (except for the environment texture);
- Ray tracing component currently supports only polygonal geometry. Points, lines and text are not supported – as a consequence, wireframe presentation cannot be rendered using ray tracing;
- Coincident triangles results in visual artifacts (same problem as in case of OpenGL usage). The workaround is to shift coincident geometries with respect to each other by a very small value;
- Support of CPUs and Intel GPUs is not tested;
- Operation under MacOS X is not tested. OCCT users are encouraged to provide their patches related to this platform;
- Source geometry for ray tracing is taken from CPU memory, existing VBO data is not reused by ray tracing. To enable OpenGL rendering in parallel with ray tracer rendering, VBOs are created without releasing CPU memory occupied by geometrical data;
- Non-optimal memory footprint: Combined (OpenGL + ray-tracing) rendering uses 3.3x memory amount (2.3x on GPU); 2 GB GPU memory provides up to 30M triangles in ray-tracing mode.



Improved presentation of markers

Since OCCT 6.7.0, there is no specific way to display some markers in a 3D scene. Instead, markers are added to a 3D scene just as a regular array of primitives (an array of points, to be precise), which is implemented as `Graphi c3d_ArrayOfPoi nts` in OCCT. See [issue 24131](#) in Visualization.

Visual attributes of the markers are defined traditionally using `Graphi c3d_AspectMarker3d` class, however this class is now taken into account by primitive array rendering code. Depending on the available OpenGL version, the markers are drawn as point sprites (OpenGL 2.0 or higher) or as bitmaps, to ensure maximum portability. Point sprite support makes markers much faster; up to 1 million markers can be drawn at acceptable frame rate even by relatively old graphic hardware (see the sample chart below).



Another improvement is that the marker image can be defined using `Image_PixMap` instance, which, in its turn, can be loaded from a raster image file or generated by an application on-the-fly.

In general, the marker image can be:

- A grey-scale image – in such a case it is considered as alpha mask which is modulated with the current marker color
- A color image – it is then used “as is” thus enabling non-monochrome markers

Management of OpenGL resources used by markers (point sprite textures or display lists) has become fully automatic. OpenGL resources are created or regenerated on demand, e.g. when a 3D view is closed and then re-opened.

`Graphi c3d_ArrayOfPoi nts` class has been extended with a possibility to specify per-vertex colors. Hence, markers can be displayed as a single array of points but using individual colors e.g. corresponding to some application-defined data value.

It should be noted, however, that even improved markers are not considered as a replacement for volume rendering techniques.

Improved presentation of dimensions

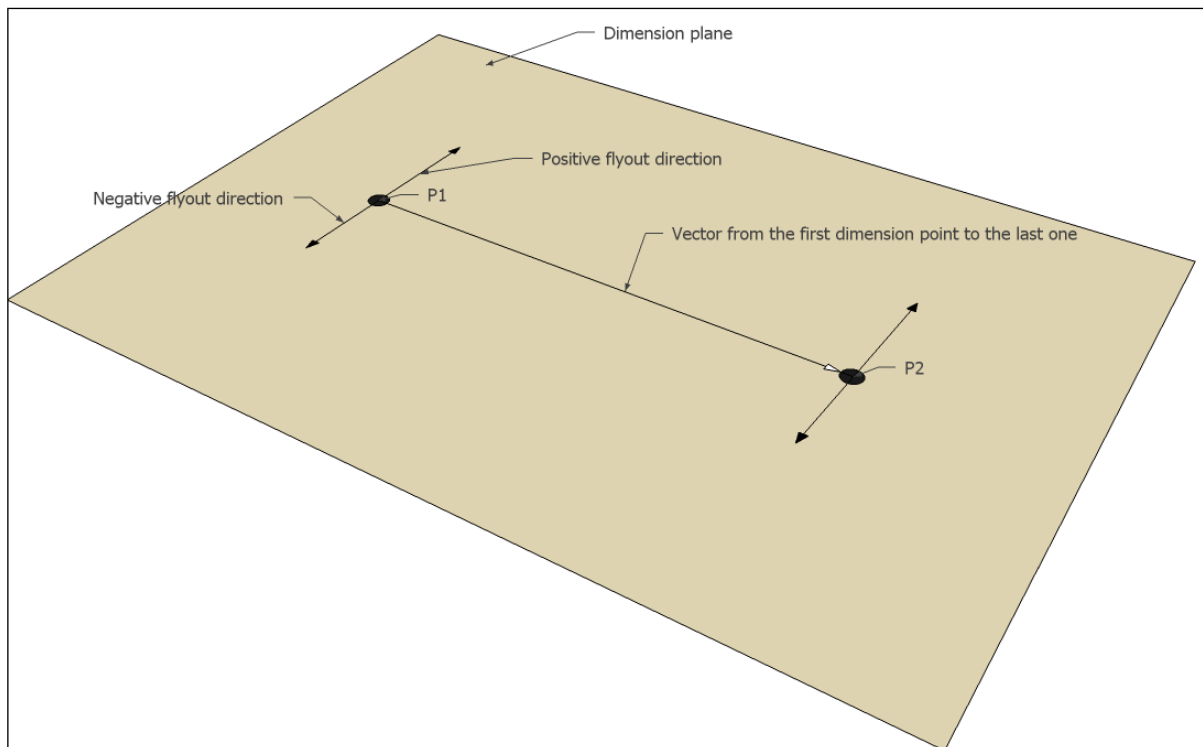
Presentation of dimensions has been redesigned and an alternative class hierarchy derived from `AIS_Dimension` on interactive object class has been introduced. See [issue 24133](#) in Visualization.

The key advantage of the new dimensions is that their geometrical parameters and appearance in a 3D view can be adjusted flexibly. The following main parameters can be tuned at the application level:

- Dimension plane
- Flyout direction and size
- Text type (2D or 3D) and position

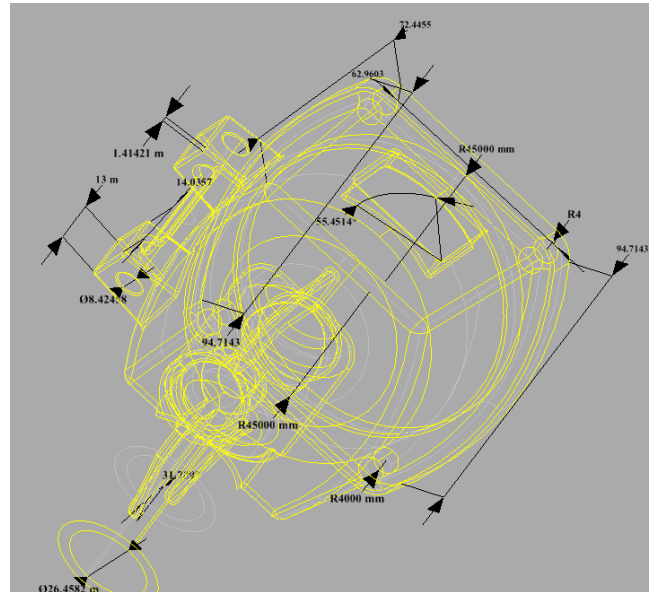
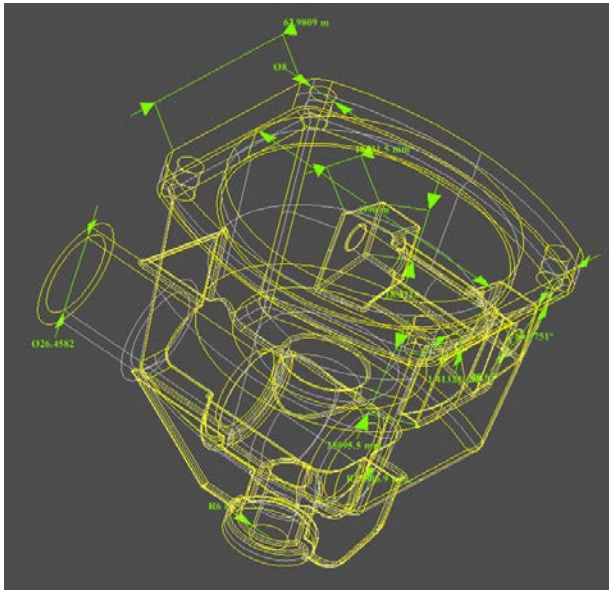
Dimension text contains an automatically computed value (length, radius, diameter or angle) thus an application should no longer care about this. Unit conversion “model units → display units” is also applied automatically, with a possibility to override both units for a given dimension instance.

Another important feature is that the new dimension classes support local selection of the main dimension elements (dimension line, dimension text). In general, the new dimensions are ready for interactive editing with the mouse at the application level.



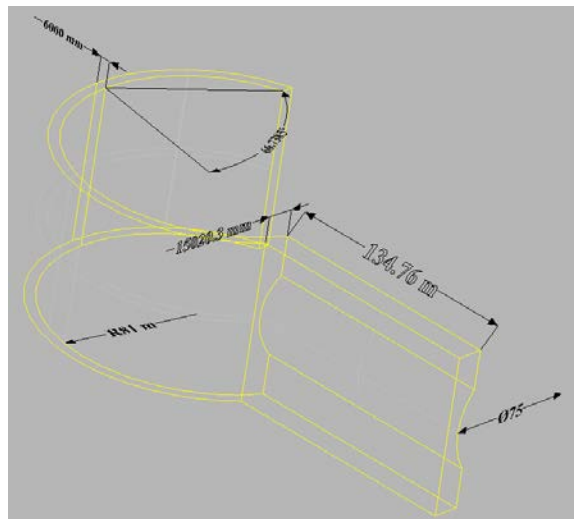
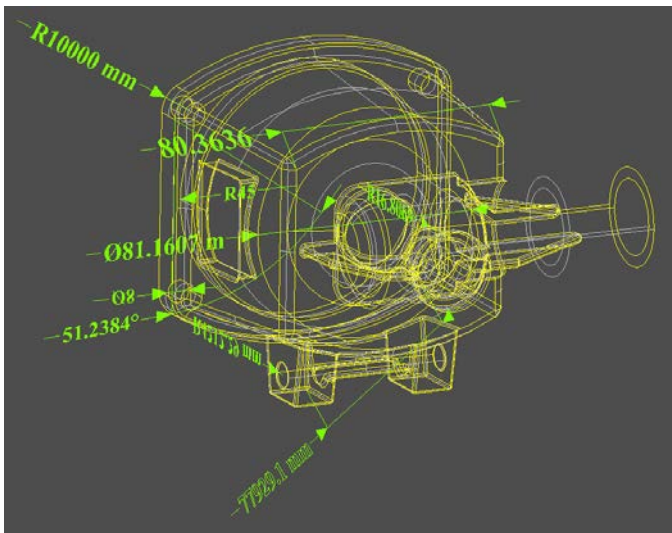
It is worth describing dimension text display options in more detail. 2D text is drawn using standard OCCT texture-mapped fonts. To make dimensions more professionally-looking, the dimension lines under the dimension labels are removed using OpenGL stencil test.





3D text is a new kind of text representation based on `Font_BRepFont` class. This class creates a `TopoDS_Shape` representing a given text string as a true 3D object in the model space. Dimensions can display 3D text using shaded or wireframe representation.

OCCT users are encouraged to evaluate this new text representation as a part of improved dimensions or as a stand-alone tool.



Old dimension classes (`AIS_Relation` and derived) are kept in OCCT 6.7.0 for compatibility reasons, however they will be removed completely in future OCCT releases.



Integration of shaders into OCCT visualization component

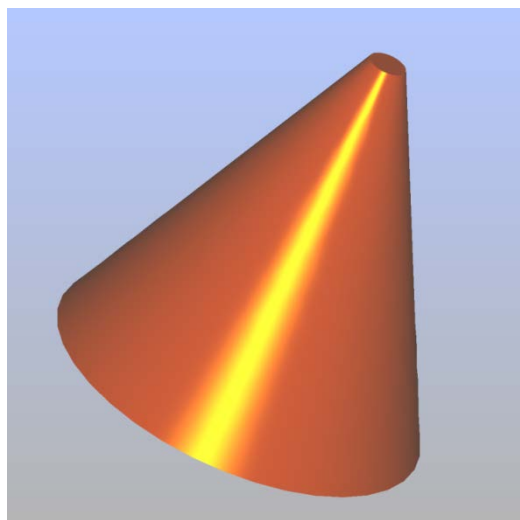
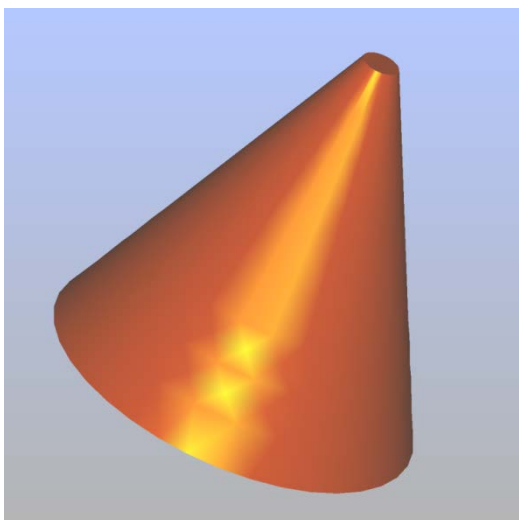
As implementation of one of the strategic steps in OCCT visualization component development road-map, support for GLSL shader programs has been added in OCCT 6.7.0. See [issue 24192](#) in Visualization.

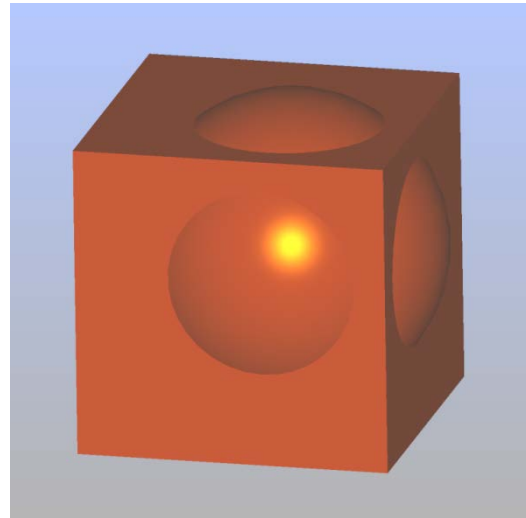
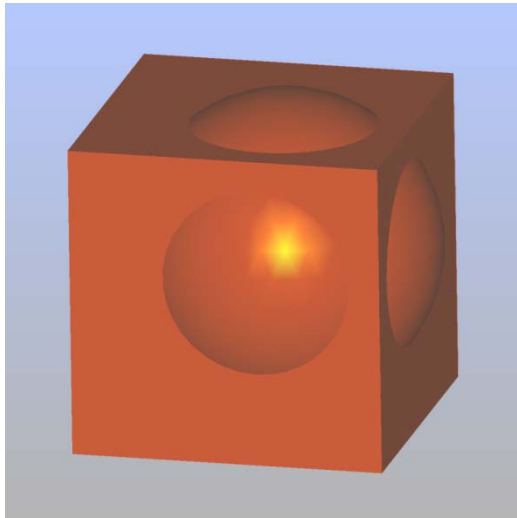
Shader programs have become a part of aspect classes in Graphic3d package, thus application developers can provide individual shader programs for all supported kinds of graphic primitives: points, lines, shaded polygons and text. Shader integration follows the general “compute, then render” paradigm:

- Application-defined implementation of `AIS_InteractiveObject::Compute()` method creates `Graphic3d_ShaderProgram` instances and passes them to `Graphic3d_Group` through the aspects, just like any other visual attributes.
- At render time, the shader programs are built and bound automatically to the OpenGL rendering context and only for the group of primitives they were associated with.

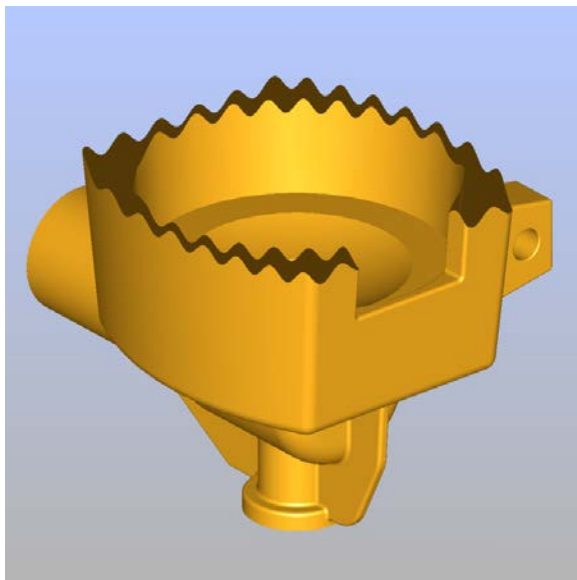
Shader management is fully automatic, just like with any other OpenGL resource. A certain limitation of the first implementation of shader support, shader programs cannot use GLSL 1.40 (or newer) capabilities. However, to overcome this limitation partially, OCCT defines a subset of uniforms needed for shader operation, so that the shader programs do not rely on GLSL 1.30 predefined variables. Refer to `Declarations.gls1` in `Shaders` directory in OCCT source tree for a complete list of uniforms defined by OCCT. In future OCCT release, redesign of the rendering logic will be continued, so as to eliminate GLSL version limitation completely.

A sample pre-defined shader program that comes with OCCT 6.7.0 implements Phong shading. It is not used by default for performance reasons. Thus it is up to an application to provide its own interactive object class that takes advantage of this shader program. Refer to `vshaderprog DRAW Test Harness` command for a usage example. The snapshots below (obtained with help of `vshaderprog` command) illustrate the difference between the default OpenGL Gouraud shading and Phong shading available in OCCT 6.7.0:

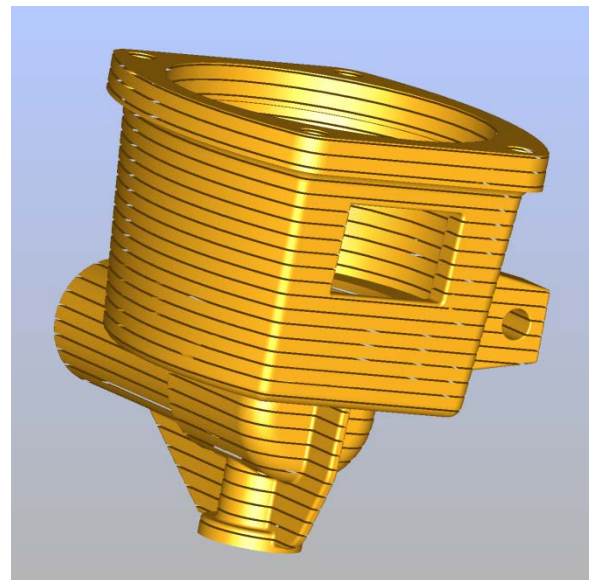




A couple of snapshots showing results of simple shader programs run by OCCT OpenGL renderer (the shader programs are not included into OCCT 6.7.0):



Fragment shader implementing custom clipping surface



Fragment shader suppressing fragments with a given step along the model Z axis

Object-level clipping planes with optional capping

Clipping visualized 3D models with user-defined clipping planes is one of basic OpenGL features that allows the user to look inside the models.

However, in some cases it is more interesting to investigate the cross-section shape rather than the model's internals. For this purpose, the cross-section should be visualized in shading using some distinctive visual attributes (specific color, hatching).

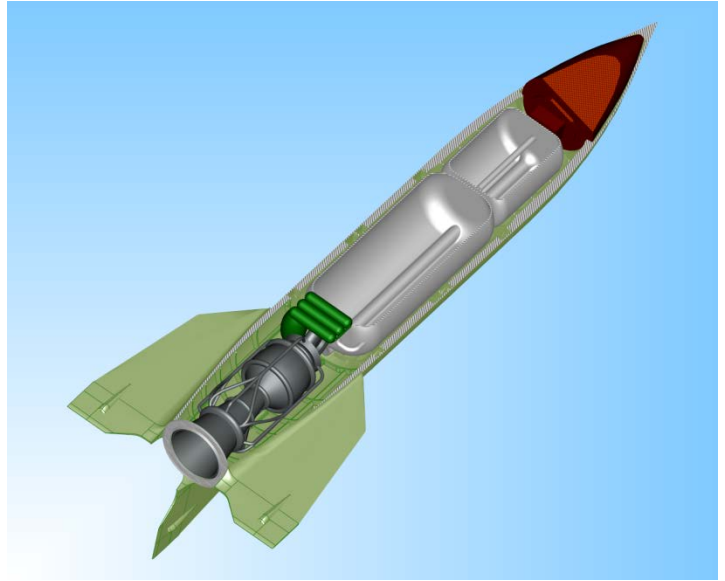
Solving this task geometrically might be a heavy work with large complicated CAD models. If visualization of the cross-section is the only goal, this task can be solved efficiently using OpenGL-based capping algorithm.



Capping algorithm is an optional property of the improved clipping plane API applicable on per-object basis. Visual attributes for cross-section are defined at the level or individual clip planes. An important consequence of this improvement is that clipping planes in general will be applicable to both the whole 3D view and individual 3D presentations.

OCCT interactive selection has also been improved to take object-specific clipping planes into account, and thus to avoid interactive selection of the object parts that were clipped.

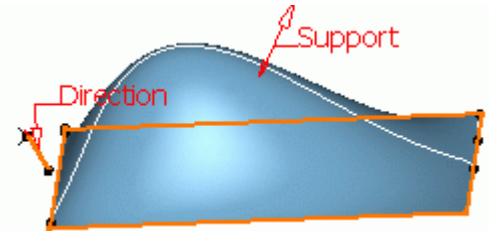
In the image to the right different parts of the rocket are clipped with different planes and hatched. See [issue 24070](#) in Visualization.



Reflection line

The reflection line on a support surface now can be built in OCCT using the new class `HLRAppli_ReflectLines`.

This is a line on the surface of the shape where normal to the surface forms an angle with the specified direction equal to 90 degrees. See [issue 23625](#) in Modeling Algorithms.



Fillet algorithms based on Newton method and analytical equations

New 2D fillet algorithm provides an easy to use interface class `ChFi2d_FilletAPI`. It constructs a fillet for any type of curves. Depending on the type it calls an analytical or an iteration algorithm. The analytical algorithm is very fast, but it works only for linear segments and arcs of circle. The iteration algorithm constructs the fillet for any type of curves, but it may be slower. Together both algorithms allow constructing a fillet for any type of curves in a fast way. See [issue 23987](#) in Modeling Algorithms.

DXF Import Interface

In Open CASCADE DXF Import interface, entities of type 'TEXT' now can be translated to shapes with help of a new class `DxfData_TranslateText` that renders text contained in a 'TEXT' entity into a set of BRep shapes. See [issue 24116](#) in Products.

The rendering of text into shapes is based on the functionality provided by new class `Font_BRepFont` from Open CASCADE Technology.

Processing of additional alignment/rotation/oblique-angle/generation-flags parameters associated with text entities, as well as replacing special symbols in text entities, is also supported in Open CASCADE DXF Import interface.

Pipe construction from a swept shell with varying section width bounded by auxiliary spine

Previously `BRepOffsetAPI_MakePipeShell` algorithm, which constructs pipes on two guide-lines, had some limitations. If the option "Keep contact" of the two guides" was set to TRUE:

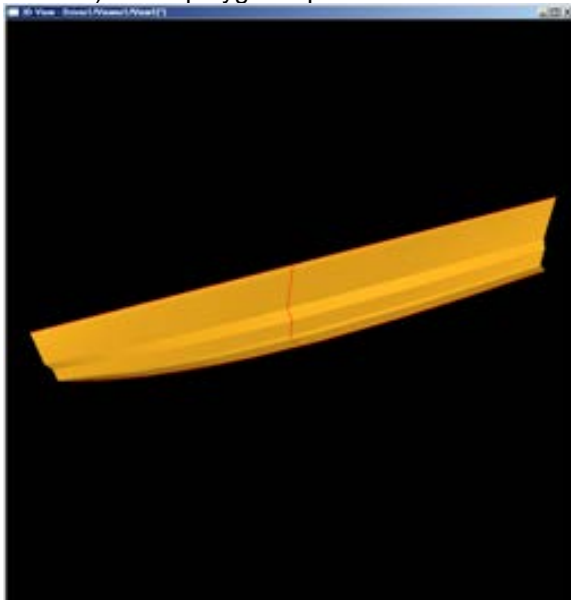
- the width of section was constant all along the path;
- the swept surface had a common point with auxiliary spine at any section.

In other words, auxiliary spine lied on the swept surface, but not necessary was a boundary of this surface. However, the auxiliary spine had to be close enough to the main spine to provide intersection with any section all along the path. If this condition was broken, the algorithm failed.

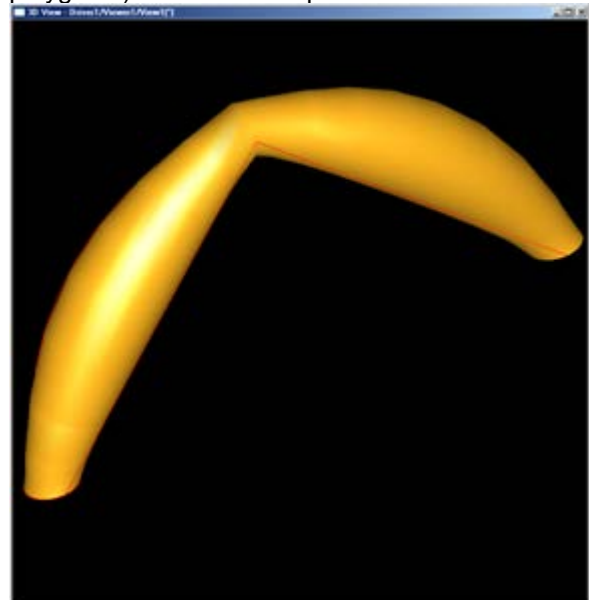
To avoid this limitation, the algorithm has been extended by a new option "Keep contact on border", which means that the auxiliary spine becomes a boundary of the swept surface and the width of section varies along the path.

This is a new type of resulting surface: the swept surface with varying width of section bounded by an auxiliary spine.

Example 1: Shell built on two guide lines (BSpline and line) and a polygonal profile



Example 2: Shell built on two guide lines (arc and polygonal) and a circular profile



Documentation

With OCCT 6.7.0 we introduce the [new documentation system](#). All OCCT documentation (Overview, User and Developer Guides) is now included in OCCT source tree (new sub-folder **doc**), in plain text format (Markdown with Doxygen extensions). The HTML, CHM, and PDF files are generated using Doxygen.

This makes it easy to make changes in OCCT documentation, and enables OCCT developers and contributors to synchronize changes in the code with corrections in relevant documentation.

Provided that you have Tcl and Doxygen installed on your system, running script **gendoc.bat** (**gendoc.sh** on Linux) located in root folder of OCCT code tree will generate HTML documentation in `doc/overview/html`.

Modifications

Foundation Classes

11758	<p><i>Summary:</i> TCollection strings are not memory safe as reported by Purify</p> <p>The classes TCollection_AsciiString, TCollection_HAsciiString and TCollection_ExtendedString have been updated to avoid use of obsolete macros defined in Standard_String.hxx.</p>
23042	<p><i>Summary:</i> Potential mistakes in (s)printf usage</p> <p>PRIuPTR macros have been implemented for formatting Standard_Size values in (s)printf family of functions, replaced by STL streams when reasonable.</p>
23843	<p><i>Summary:</i> scanf without field width limits can crash with huge input data.</p> <p>The following improvements have been introduced in code using *printf and *scanf functions:</p> <ul style="list-style-type: none"> ▪ Width has been correctly indicated; ▪ Use of buffer arrays has been improved; ▪ Unreferenced variable and other compiler warnings have been eliminated.
23919	<p><i>Summary:</i> Redefine operator* for Handle_Standard_{Transient Persistent} subclasses</p> <p>The operator* has been redefined in Standard_DefineHandle template macros to reduce the code and make OCC handle more compliant with NCollection_Handle and std::shared_ptr.</p>
23920	<p><i>Summary:</i> Use of static variables in Message package</p> <p>Previously Message_MsgFile class used static variables to create messages for undefined keywords. When running Shape Healing concurrently this lead to data races when the message files have not been loaded upfront (e.g. due to failure to set environment variables CSF_SHMessage in the end-user environment).</p> <p>To minimize the risk of data races when running Shape Healing concurrently, the static variables for error messages are now set only once. This gives a possibility to enforce calling the method Message_MsgFile::Msg() upfront to initialize these variables and read-access them afterwards.</p>
23949 24030	<p><i>Summary:</i> The size of the second and further blocks allocated by OCC IncAllocator differs from the size of the first one</p> <p>NCollection_IncAllocator has been tuned to allocate blocks of equal size to reduce fragmentation and increase chances for block reuse.</p>
23988	<p><i>Summary:</i> Force use of reentrant mode</p> <p>The reentrant mode is now always on and cannot be switched off. To implement this:</p> <ul style="list-style-type: none"> ▪ Methods IsReentrant and SetReentrant have been removed from package Standard, ▪ Allocators have been corrected in Standard and NCollection. ▪ Use of IsReentrant and SetReentrant has been eliminated in Products.

<p>24042 24044 24072</p>	<p><i>Summary:</i> Performance improvements</p> <p>The following improvements have been introduced to speed up computation of intersection of two surfaces:</p> <ul style="list-style-type: none"> ▪ <code>TCollection_Array1::Init()</code> has been replaced with <code>[i]</code> to improve compiler vectorization and provide consistency with <code>Array2::Init()</code>; ▪ New method <code>NCollection_BaseCollection::Allocator()</code> returns allocator of the collection, which is STL-consistent and can be reused; ▪ Automatic allocation (on stack) has been implemented in <code>math_DoubleTab</code> and <code>math_SingleTab</code> classes to minimize heap allocation. Size constants have been selected to avoid stack overflow but maximize chances for stack allocation.
<p>24137</p>	<p><i>Summary:</i> <code>math_FunctionSetRoot</code> returns too rough solution</p> <p>The range for <code>DescentIter</code> parameter from <code>math_FunctionSetRoot</code> class has been increased from 10 to 100 to handle rare cases with better precision.</p>
<p>24191</p>	<p><i>Summary:</i> Static assert functionality</p> <p><code>Standard_STATIC_ASSERT</code> macro has been added for compile-time asserts on pre-C++11 compilers. The new macro is used in <code>Standard_MgrOpt</code> and <code>QANCollection</code> classes.</p>
<p>24259</p>	<p><i>Summary:</i> Clean up <code>OSD_File</code> implementation</p> <p>The following changes have been introduced in <code>OSD_File</code> implementation:</p> <ul style="list-style-type: none"> ▪ Field <code>myFileChannel</code> has been removed from <code>OSD_FileNode</code> and introduced to <code>OSD_File</code>. ▪ <code>OSD_File</code> implementation has been cleaned from redundant type casts; some missing type cases have been added. ▪ Method <code>OSD_File::Size()</code> now returns <code>Standard_Size</code>, which provides correct value for large files (on 64-bit platforms only).
<p>24271</p>	<p><i>Summary:</i> Provide Boolean operations for <code>NCollection_Map</code></p> <p>Implementation for Boolean operations Union, Intersection, Subtraction and Difference has been added in <code>NCollection_Map</code> class. Method <code>Exchange()</code> has been added in <code>NCollection</code> map classes to allow to efficiently exchanging data contained in two map instances without unnecessary allocations and copy operations.</p>
<p>24300</p>	<p><i>Summary:</i> Remove unused class <code>MMgt_StackManager</code></p> <p>Unused class <code>MMgt_StackManager</code> has been removed from OCCT.</p>
<p>24380</p>	<p><i>Summary:</i> Improvement to follow the maximum peak size</p> <p>New parameter <code>myBreakPeak</code> has been introduced in <code>OSD_MailocHook</code> to set a user defined peak size limit to debug.</p>



Application Framework

3513	<p><i>Summary:</i> Check for array boundary in method Set for array attributes</p> <p>Check for array boundaries has been implemented in methods Set(i, val) from packages TDataStd and BOPDS.</p>
23766	<p><i>Summary:</i> Redesign mechanism of shape orientation processing</p> <p>The mechanism of shape orientation processing has been redesigned in packages PNaming and TNaming MNaming, MDF, BinMNaming, BinLDri vers, Xml MNaming and Xml LDri vers.</p> <p>Previously the orientation was stored in the document in an extra field of Selection method. Now orientation of the shape is stored in the dedicated field myOrientation of TNaming_Name class. The compatibility with previous document formats is preserved.</p>
23850	<p><i>Summary:</i> TdataStd_ByteArray is too slow on storage on disk</p> <p>The following modifications have been introduced to optimize storage of OCAF data on disk:</p> <ul style="list-style-type: none"> ▪ XML persistence of TDataStd_ByteArray and TDataStd_TreeNode has been refactored. ▪ Storage of some OCAF attributes in XML file format has been accelerated. ▪ Format of storage of a double value is extended to keep 17 digits after a decimal point (only 15 digits were used previously). ▪ Plib_LocalArray has been renamed into Ncollection_LocalArray and has become a template. It is used as a local array for Standard_Character in XML OCAF drivers, and as a local array of Standard_Real in Plib package.
23864	<p><i>Summary:</i> & symbol is read incorrectly from a XML Ocaf file</p> <p>& symbol support has been restored in method LDOM_CharReference::Decode.</p>
23912	<p><i>Summary:</i> Optimization of TdataStd_ExtStringArray::Value()</p> <p>TdataStd_ExtStringArray::Value() has been modified to return a (constant) reference to the string value.</p>
23935	<p><i>Summary:</i> Compiler warnings on returning Handle from C functions in OCAF schemas</p> <p>Create##schema method has been removed from Storage_Macros.hxx. It is possible to use new##Schema method instead.</p>
24164	<p><i>Summary:</i> Optimization of OCAF document closing</p> <p>OCAF document closing has been optimized. New method TDocStd_Document::BeforeClose allows preparing the document for closing.</p>



24172 24212	<p><i>Summary:</i> Compiler warnings <code>unreacheable code</code> and <code>unreferenced local function</code> in OCAF</p> <p>Cycles <code>for()</code> with body always ending with <code>break</code> have been replaced by equivalent <code>if()</code> statements. <code>Sample</code> files containing unused sample code have been removed from several OCAF packages and put in the <code>samples/ocafsamples</code> folder.</p>
24263	<p><i>Summary:</i> <code>TNaming_CopyShape::CopyTool</code> failure</p> <p><code>TNaming_CopyShape::CopyTool</code> method now can be used directly (without <code>TNaming_Translator</code>).</p>
24280	<p><i>Summary:</i> Behavior of <code>TObj_Model::GetFile()</code> is incorrect</p> <p><code>TObj_Model::GetFile</code> function has been modified to avoid throwing the exception if the model was not stored yet.</p>

Modeling Data

23863	<p><i>Summary:</i> Wrong distance value between circle and cylinder</p> <p>The method <code>Extrema_ExtElCS::Perform</code>, which calculates the minimum distance between a circle and a cylinder, now previously checks their intersection.</p>
23939	<p><i>Summary:</i> Incorrect circle parameter in <code>IntAna</code></p> <p>The parameters of intersection between two circles in case of one tangency point have been corrected in method <code>IntAna2d_AnaIntersection::Perform</code>.</p>
23945	<p><i>Summary:</i> <code>GeomAdaptor_Surface</code> fails to compute the first derivatives on the surface of the attached face</p> <p><code>FKIIdx</code> and <code>LKIIdx</code> parameters are now used in <code>GeomAdaptor_Surface</code> algorithm instead of <code>NbKnots</code>.</p>
23964	<p><i>Summary:</i> <code>Extrema_ExtXX::Point</code> methods should return constant reference instead of copy</p> <p>Method <code>Point(const Standard_Integer N)</code> now returns constant reference instead of copy in the following classes: <code>Extrema_ExtPElC</code>, <code>Extrema_ExtPElC2d</code>, <code>Extrema_ExtPElS</code>, <code>Extrema_ExtPExtS</code>, <code>Extrema_ExtPRevS</code>, <code>Extrema_ExtPS</code> and <code>Extrema_ExtPSofRev</code>.</p>
24028	<p><i>Summary:</i> It is impossible to create a face based on <code>Geom_RectangularTrimmedSurface</code> by standard methods</p> <p>Trimmed Surfaces have been replaced with bases in IGES and STEP import; <code>BRepLib_MakeFace</code> now uses the <code>trimmed</code> surface (instead of base surface) if <code>Geom_RectangularTrimmedSurface</code> is given for the face creation.</p>
24032	<p><i>Summary:</i> An exception raised during projection of the curve on the surface</p> <p>Method <code>ProjLib_ComputeApproxOnPol arSurface::Handle</code> has been modified to make the projection of a curve on surface more precise in some cases.</p>



24062	<p><i>Summary:</i> A mistake in <code>Geom_OffsetCurve.cdl</code> about direction of offset</p> <p>A misprint in description has been fixed in <code>Geom_OffsetCurve.cdl</code>.</p>
24076	<p><i>Summary:</i> New warning during compilation OCCT on WNT platform</p> <p>Return statement has been added in method <code>Adaptor3d_IsoCurve::Ellipse</code> to disable warning.</p>
24081	<p><i>Summary:</i> Memory corruption when projecting point on surface of revolution</p> <p>The method <code>Extrema_ExtPREvS</code> now provides allocation for up to 8 solutions and uses 0-based indexing. Earlier it provided up to 6 solutions and used 1-based indexing, which caused memory corruption.</p>

Modeling Algorithms

22766	<p><i>Summary:</i> Wrong results of intersection between a surface of revolution and a plane</p> <p>Precision check has been added in function <code>MyDirFunction::Value</code> from <code>math_FunctionSetRoot</code> class.</p>
23360	<p><i>Summary:</i> Test cases for command <code>mkoffset</code> produce different results on different versions of OCCT</p> <p>The following improvements have been introduced in the offset algorithm:</p> <ul style="list-style-type: none"> ▪ Function <code>QuasiFleche()</code> from class <code>BRepFill_OffsetWire</code> has been protected against calling for too small edges; ▪ Function <code>OffsetWire::FixHoles()</code> has been protected against NULL-shapes; ▪ Function <code>ProjOnCurve()</code> from <code>Bisector_BisecCC</code> class now returns status (done or not done).
23582	<p><i>Summary:</i> Argument of <code>LocOppe_WiresOnShape::SetCheckInterior()</code> function is ignored.</p> <p>Method <code>LocOppe_WiresOnShape::SetCheckInterior()</code> has been corrected to properly use argument <code>ToCheckInterior</code>.</p>
23782	<p><i>Summary:</i> Intersection algorithm produces wrong section curves for the attached faces</p> <p>The method <code>Adaptor3d_TopolTool::BsplSamplePnts</code> has been modified to preserve the correlation between numbers of segments in U and V parametric directions (<code>nbu</code> and <code>nbv</code>).</p> <p>When one of these numbers (e.g. <code>nbu</code>) is set to the default minimal value (10), the other (<code>nbv</code>) is set to $\text{Min}(30, \text{nbv} * (\text{default} / \text{nbu}))$. The value 30 is used to avoid too small segments.</p>



<p>23625</p>	<p><i>Summary:</i> Reflection lines on a surface</p> <p>The reflection line on a support surface is a new type of curve that can now be built in OCCT. This is a line on the surface of the shape where normal to the surface forms an angle with the specified direction equal to 90 degrees.</p> <p>The reflection line is implemented in <code>HLRAppli_ReflectLines</code> class and can be defined by the following input parameters:</p> <ul style="list-style-type: none"> ▪ Support shape (face or shell) with a convex or concave underlying surface that also can contain holes; ▪ Direction (axis or straight line); <p>Note that input shapes should be valid according to OCCT validity criteria.</p> <p>The result is an Edge (or Compound of Edges, i.e. multi-result).</p> <ul style="list-style-type: none"> ▪ If the computed reflection line is at the back side of the input shape it will be removed from the result. ▪ If the computed line has a 'loop' it will be interpreted as two separate lines having one touch point.
<p>23650</p>	<p><i>Summary:</i> Slow meshing of a bspline surface</p> <p>The algorithms of normal computation have been corrected in method <code>GeomLib::NormEstim</code>.</p>
<p>23706</p>	<p><i>Summary:</i> Cannot project point on curve</p> <p>The following improvements have been introduced in processing of curves and surfaces with singularity points:</p> <ul style="list-style-type: none"> ▪ Approximation by Taylor-series and by three points has been implemented for computation of tangents in singularity points. Expansion in a Taylor-series is used to compute the tangent vector if it exists in a singularity point while the curve derivative in this point is equal to zero. Approximation by three points is required if the first three derivatives in a singularity point are equal to zero and only the fourth derivative is not. ▪ <code>GetNormal()</code> method has been added in class <code>gp_Vec2d</code>. ▪ Handling of Singularity point has been introduced in the algorithm computing Value, D0, D1, D2 and D3 for offset curves in classes <code>Geom_OffsetCurve</code> and <code>Geom2d_OffsetCurve</code>. ▪ New method <code>GeomFillFrenet::RotateTrihedron</code> revolves the trihedron (defined by "Tangent", "Normal" and "BiNormal" vectors) to coincide "Tangent" and "NewTangent" axes. ▪ Restriction on the <code>LastParameter</code> for visualization of 3D curves has been implemented in <code>DrawTrSurf_Drawable::DrawCurveOn()</code> function. <code>PlotCurve()</code> function is now called for the last interval. ▪ The classes <code>Lprop_CLProps</code> and <code>Lprop_SLProps</code> have been modified to provide tangent computing in a singularity point.
<p>23708</p>	<p><i>Summary:</i> <code>BRepCheck</code> considers a correct shape as invalid.</p> <p><code>BRepCheck</code> algorithm has been modified to avoid considering a shape invalid if its external and internal wires have one common vertex.</p>
<p>23839</p>	<p><i>Summary:</i> Projection algorithm produces wrong results for a set of data</p> <p>Projection algorithm has been improved in class <code>ProjLib_ComputeApproxOnPol arSurface</code>.</p>





<p>23906 24069</p>	<p><i>Summary:</i> Performance of the projection algorithm became worse in some cases</p> <p><code>Extrema_ExtPS</code> that is used in the algorithm of a point on surface projection in Boolean operations has been set to search for only minimal distance.</p>
<p>23901</p>	<p><i>Summary:</i> Memory leaks in tests</p> <p>Memory leaks caused by cyclic smart pointers have been fixed in BOPDS package.</p>
<p>23903</p>	<p><i>Summary:</i> Invalid result of pipe operation on closed path</p> <p>BRepFill_Sweep algorithm has been corrected to provide proper result of pipe operation on closed path.</p>
<p>23933</p>	<p><i>Summary:</i> Self intersection reported after Fuse operation.</p> <p>The check to avoid creating an Edge/Face intersection if the edge lies on the surface of the face, but not on the face itself, has been added in <code>IntTools_BeanFaceIntersection::FastComputeExactIntersection()</code>.</p>
<p>23943</p>	<p><i>Summary:</i> OCC fails to work with offset surfaces with singularities</p> <p>Unlike other types of surfaces, offsets throw an exception <code>Geom_UndefinedValue</code> (or <code>Geom_UndefinedDerivative</code>) when trying to compute values near their singularities. The algorithms that do not expect that surface evaluation throws an exception eventually produce incorrect results when working with such geometries.</p> <p>The fix addresses two cases trying to improve robustness of OCC:</p> <ul style="list-style-type: none"> ▪ Singularity is now recognized in <code>Geom_OsculatingSurface</code> if the derivative is below the tolerance (regardless that it should be greater than 1e-12 threshold used before) ▪ <code>ShapeAnalysis_Surface::ValueOf()</code> now avoids going beyond the surface range for offset surfaces.
<p>23952</p>	<p><i>Summary:</i> Thread-safety of intersections, approximations and other modeling algorithms</p> <p>Multiple modifications have been introduced to improve thread-safety of core classes used in surface-surface intersections and surface approximations, which consequently improves the thread-safety of such algorithms as sweeps, etc.</p>
<p>23958</p>	<p><i>Summary:</i> Section of shell by plane is incomplete</p> <p>The algorithm of intersection of planar surfaces has been improved to provide correct result in complex cases.</p>
<p>23981</p>	<p><i>Summary:</i> Wrong intersection curves</p> <p>The following improvements have been introduced in the intersection algorithm:</p> <ul style="list-style-type: none"> ▪ Processing of infinite surface boundaries has been corrected in <code>GeomInt_IntSS</code> class. ▪ The algorithm of point projection on periodic surfaces has been fixed to avoid wrong periodic adjusting of the solution.





<p>23982</p>	<p><i>Summary:</i> Wire explorer raises exception</p> <p>The method <code>BRepTools_WireExplorer::Next()</code> has been modified to provide correct processing of edges with zero length.</p>
<p>23985</p>	<p><i>Summary:</i> There is no section between attached faces</p> <p>New parameter "H", the height of the cylinder, has been added to classes <code>IntAna_IntConicQuad</code> and <code>IntAna_QuadQuadGeo</code>. This parameter is used for checking if the plane and cylinder are parallel in case when the angle between the cylinder axis and the plane normal is very close to π.</p> <p>If the "height" is so big that a point translated from the point of intersection between the cylinder axis and the plane by the distance "height" in the direction of cylinder axis does not belong to the plane, this plane and the cylinder cannot be considered parallel.</p>
<p>23987</p>	<p><i>Summary:</i> 2D fillets and chamfers for any type of edges in a plane</p> <p>The following new algorithms for creation of 2D fillets and chamfers have been implemented in <code>ChFi2d</code> package:</p> <ul style="list-style-type: none"> ▪ <code>ChFi2d_Builder</code> constructs a fillet or chamfer for linear and circular edges of a face. ▪ <code>ChFi2d_AnafilletAlgo</code> is an analytical constructor of the fillet. It works only for linear and circular edges, having a common point. ▪ <code>ChFi2d_FilletAlgo</code> is an iteration recursive method constructing the fillet edge for any type of edges including ellipses and b-splines. The edges may even have no common points. ▪ <code>ChFi2d_ChamferAPI</code> constructs of chamfers between two linear edges of a plane. <p>The algorithms <code>ChFi2d_AnafilletAlgo</code> and <code>ChFi2d_FilletAlgo</code> may be used directly or via the interface class <code>ChFi2d_FilletAPI</code>.</p>
<p>23994</p>	<p><i>Summary:</i> <code>GeomAPI_ExtremaCurveCurve</code> class calculates wrong values</p> <p>The new sampling algorithm has been added in class <code>Extrema_CurveCache</code> for <code>Bspline</code> curves with degree ≤ 3. This algorithm divides the curve uniformly between knots. In some cases it performs better than the old uniform sampling along a parameter.</p>
<p>23995</p>	<p><i>Summary:</i> <code>GeomAPI_ExtremaCurveSurface</code>: wrong result between a curve and a plane</p> <p>The algorithm of search of min and max distances on C^0-continuous curves has been corrected in <code>Extrema_ExtCS</code> class.</p>
<p>23998 24140 24360</p>	<p><i>Summary:</i> Improved algorithm of intersection of two faces</p> <p>Method <code>IntWalk_Pwalking::TestArret</code> has been modified to avoid loop on curve to check if the boundaries are not overrun at the solution point (instead of the check in the current initial point as previously).</p>





24004	<p><i>Summary:</i> Initialization of arrays <code>Tpoints</code>, <code>Tedges</code> and <code>Ttriangles</code> of the class <code>IntPol yh_Mai l l ageAffi nage</code> by exact values</p> <p>The arrays <code>Tpoints</code>, <code>Tedges</code> and <code>Ttriangles</code> of the class <code>IntPol yh_Mai l l ageAffi nage</code> are now initialized by exact values; Loops in the class <code>IntPol yh_Mai l l ageAffi nage</code> have been optimized.</p>
24005	<p><i>Summary:</i> Intersecting a slightly off angle plane with a cylinder takes 7+ seconds</p> <p>If the new algorithm of intersection between plane and cylinder produces an ellipse that is too huge to be efficiently processed, the necessary part of the ellipse is now approximated by B-spline using the old intersection algorithm.</p> <p>The corresponding changes have been introduced in <code>IntPatch</code>, <code>IntTools</code> and other packages.</p>
24016	<p><i>Summary:</i> Test case <code>moddata_2 bug324</code> does not work correctly</p> <p>The method <code>IntPatch_ImpImpIntersection::Perform</code> has been fixed to avoid checking infinite cylinders for the height of the cylinder.</p>
24025	<p><i>Summary:</i> <code>BRepAl goAPI_Boo l eanOperati on::Modi fi ed2()</code> should be removed</p> <p>The obsolete method <code>BRepAl goAPI_Boo l eanOperati on::Modi fi ed2()</code> has been removed (in the previous version of BOP it used History collector to find modified shapes, but there is no such collector in current version of BOP).</p> <p>The replacement for this method is <code>BRepAl goAPI_Boo l eanOperati on::Modi fi ed()</code>.</p>
24033	<p><i>Summary:</i> Orientation of all shapes is set to <code>INTERNAL</code> as a result of <code>BRepAl goAPI_Co mmon</code></p> <p>Dimension of the shapes is now checked when choosing the member shape for common operation in function <code>BOPAl go_BOP::Bui l dRC</code>.</p>
24035	<p><i>Summary:</i> <code>Intersector</code> is not symmetrical</p> <p>Function <code>IntCurve_IntPol yPol yGen::Fi ndI ntersect</code> has been corrected to ensure that the check of intersection is symmetrical.</p> <p>The calculation of parameter on curve for the last polygon segment has been corrected.</p>
24036	<p><i>Summary:</i> Regression: sewing is not correct</p> <p>The algorithm of finding candidates for sewing has been improved in <code>BRepBui l derAPI_Sewi ng</code> class.</p>
24037	<p><i>Summary:</i> Wrong result done by General Fuse algorithm</p> <p>The algorithm finding angle between same edges in function <code>BOPAl go_Wi reSpl i tter::Path</code> has been improved.</p>



24040	<p><i>Summary:</i> The result of CUT operation is not correct</p> <p>The algorithm finding a point inside face to detect same domain faces has been improved in method <code>BOPTools_AlgoTools::AreFacesSameDomain</code>.</p>
24043	<p><i>Summary:</i> Performance improvements</p> <p>The following performance improvements have been introduced:</p> <ul style="list-style-type: none"> ▪ Inheritance from <code>gp_Storable</code> has been removed from class <code>Extrema_PonSurf</code> to reduce memory footprint; ▪ More efficient containers and allocators have been implemented in <code>GeomInt</code> package; ▪ The use of <code>sqrt()</code> has been reduced in <code>IntPatch</code> package; ▪ The use of <code>Ncollection_IncAllocator</code> has been reduced in <code>IntSurf</code> package; ▪ Caching of loop variables is avoided in <code>Intf</code> package; ▪ Copy construction of <code>ThePathPoint</code> is avoided in <code>IntWalk_Iwalking</code> class; ▪ <code>Ncollection_IncAllocator</code> has been implemented in <code>IntWalk_Iwalking</code> class.
24053	<p><i>Summary:</i> Section between plane and sphere is not correct</p> <p>Splitting into intervals has been corrected in method <code>IntTools_FaceFace::SetList</code>.</p>
24060	<p><i>Summary:</i> Wrong result done by general fuse algorithm</p> <p>Processing of vertex as a single object has been added in <code>BOPAlgo_PaveFiller</code> class.</p>
24068	<p><i>Summary:</i> Wrong result done by projection algorithm</p> <p>The number of nodes needed to create a tree for <code>BsplineSurface</code> has been corrected in method <code>Extrema_GenExtPS::BuildTree</code> to be not less than the degree multiplied by knots number.</p>
24075	<p><i>Summary:</i> Boolean Section between two faces fails</p> <p>A misprint has been corrected in method <code>IntTools_FaceFace::ComputeTolReached3d()</code>.</p>
24083	<p><i>Summary:</i> <code>BRepCheck_Wire</code> crashes processing a wire consisting of many edges</p> <p>A recursive method <code>Propagate()</code> has been replaced by a non-recursive implementation in class <code>BRepCheck_Wire</code>.</p>
24089	<p><i>Summary:</i> Missing section edge</p> <p>Method <code>BOPDS_DS::VerticesOnIn</code> has been improved to add to the resulting map the vertices that are On/In for both faces.</p>
24092	<p><i>Summary:</i> Boolean fuse fails</p> <p>Method <code>IntTools_FaceFace::SetList</code> has been fixed to obtain correct results for a fuse of two faces.</p>

24193	<p><i>Summary:</i> Useless iterations in <code>IntStart_SearchOnBoundaries_1.gxx</code></p> <p>The function <code>IntStart_SearchOnBoundaries::BoundedArc()</code> has been modified to avoid useless iterations.</p>
24098	<p><i>Summary:</i> Exception <code>Standard_OutOfMemory</code> is raised during topological operation.</p> <p>The following improvements have been introduced to improve memory management of Boolean Operations:</p> <ul style="list-style-type: none"> ▪ New method <code>BOPDS_Tools::TypeToInteger</code> converts the type of shape <code><theT></code> to the integer value, that is returned. ▪ Method <code>BOPDS_DS::Init()</code> has been modified to avoid adding a shape argument in the data structure if it already exists there. ▪ The intersection pairs containing a shape and sub-shape of the shape are not taken into consideration by method <code>BOPDS_Iterator::Intersect()</code>. ▪ The memory allocators in method <code>BOPAI_go_Builder::FillIn3Dparts</code> have been shifted to local scopes to release the memory as soon as possible.
24101	<p><i>Summary:</i> Self-Interference checker provides incorrect results</p> <p>The following modifications have been introduced to improve self-Interference checker:</p> <ul style="list-style-type: none"> ▪ The check for tangency of surfaces has been implemented via new methods <code>BOPDS_InterfFF::SetTangentFaces</code>, which sets the flag if the faces are tangent, and <code>BOPDS_InterfFF::TangentFaces</code>, which returns this flag. ▪ The information about tangency is taken into account by method <code>BOPAI_go_PaveFiller::PerformFF()</code> that checks face-face interference.
24106	<p><i>Summary:</i> Regression in <code>PointNearEdge</code></p> <p>Threshold value for a step in 2D-space now correctly takes into account 2D-bounds of the face in method <code>BOPTools_AlgoTools3D::PointNearEdge</code>.</p>
24107	<p><i>Summary:</i> <code>BrepAIgo::ConcatenateWireCO</code> method doesn't work on a translated wire</p> <p>The algorithm of wire concatenation has been improved in <code>BrepAIgo</code> class.</p>
24108	<p><i>Summary:</i> Boolean fuse fails</p> <p>The tolerance of checking the distance between a vertex and an edge in <code>BOPoint_Context::ComputeVE()</code> has been increased by additional <code>Precision::Confusion()</code> to detect the interference.</p>
24112	<p><i>Summary:</i> Unused variables initialized in <code>Extrema</code> package</p> <p>Unused variables have been removed from several classes of <code>Extrema</code> package.</p>
24122	<p><i>Summary:</i> Hang-up during a topological operation.</p> <p>An unbalanced binary tree has been implemented in method <code>BOPAI_go_Builder::FillIn3Dparts</code> to improve the performance.</p> <p>Data processing for tangent cases has been changed in method <code>BOPTools_AlgoTools::GetFaceOff</code>.</p>



24134	<p><i>Summary:</i> Wrong result of projection point on the face</p> <p><code>Bnd_SphereUBTreeSelector</code> has been modified to provide correct work of <code>Extrema_ExtAlgo_Tree</code> projection algorithm.</p>
24138	<p><i>Summary:</i> Exception during projection of the point on the face</p> <p>The algorithm of point projection on a surface of extrusion in class <code>Extrema_ExtPExtS::Perform</code> has been corrected to avoid generation of wrong solutions.</p>
24142	<p><i>Summary:</i> Wrong section curve</p> <p>In case of intersection of two cones, which have almost the same axis of revolution, the vectors are now normalized before projecting the apex of one cone to the axis of revolution of another cone. This algorithm is provided by the class <code>IntAna_QuadQuadGeo</code>.</p>
24143	<p><i>Summary:</i> bopcheck command throws an exception</p> <p><code>IntTools_BeanFaceIntersector::ComputeLocalized()</code> now uses the tolerance value of the face instead of the sum of face and edge tolerance values to compute grid points on the face surface.</p>
24144	<p><i>Summary:</i> <code>BRepPrimAPI_MakeWedge</code> wrong</p> <p>The algorithm computing direction has been fixed in method <code>Primitives_Wedge::Line</code>.</p>
24154	<p><i>Summary:</i> Wrong result of CUT operation</p> <p>New method <code>BOPTools_AlgoTools::GetFaceDir</code> allows rebuilding the normal to a face if the method <code>FindPointInFace</code> is unable to find point in the face in bi-normal direction.</p>
24167 24222	<p><i>Summary:</i> Compiler warnings 'unreacheable code' and 'conditional expression is constant'</p> <p>The code has been revised to get rid of MSVC compiler warnings C4702: "unreachable code" and C4127: "conditional expression is constant".</p>
24174	<p><i>Summary:</i> <code>BRepBulderAPI_Sewing</code> returns result with very high tolerance</p> <p>Precision parameter has been corrected in method <code>BRepBulderAPI_Sewing::SameParameterEdge</code>.</p>
24187	<p><i>Summary:</i> Wrong result of COMMON operation</p> <p>Processing of arguments of type <code>COMPOUND</code> has been added in method <code>BOPAlgo_Bulder::FillInternalShapes</code>.</p>
24190	<p><i>Summary:</i> Exception raised during topological operation</p> <p>"Try/catch" block has been added in method <code>BOPAlgo_CheckerSI::Perform</code>.</p>





24195	<p><i>Summary:</i> Use of uninitialized data in <code>IntCurve_IntPolyPolyGen.gxx</code></p> <p>The situation when the intersection algorithm cannot find a solution is now correctly processed by method <code>IntCurve_IntPolyPolyGen::FindIntersect()</code>.</p>
24200	<p><i>Summary:</i> Wrong result obtained by <code>Extrema</code> during Curve/Curve intersection</p> <p>The number of nodes in case of <code>GeomAbs_Line</code> in <code>Extrema_GextCC::Perform()</code> has been changed.</p> <p>Coefficient check has been added in <code>Extrema_CurveCache::Extrema_CurveCache</code> to avoid integer overflow.</p>
24213	<p><i>Summary:</i> <code>bopargcheck</code> complains on sphere</p> <p>The degenerated edges are now correctly checked in class <code>BOPAlgo_ArgumentAnalyzer</code>.</p>
24220	<p><i>Summary:</i> <code>bopargcheck</code> returns valid for C0 shape but results of Boolean operations are broken with such shapes</p> <p>The command <code>bopargcheck</code>, which detects problems affecting Boolean operations, now also detects the shapes with 0D geometries.</p>
24235	<p><i>Summary:</i> <code>BRepBuilderAPI_Sewing</code> – add protection against too small tolerance</p> <p>Method <code>BRepBuilderAPI_Sewing::Init</code> has been protected against too small tolerance.</p>
24242	<p><i>Summary:</i> Hang-up during classification of a 3D point relative to a solid</p> <p>The method <code>BrepClassifier3d_SolidExplorer::OtherSegment</code> has been modified to avoid infinite loop.</p>
24247	<p><i>Summary:</i> Wrong result obtained by General Fuse algorithm</p> <p>The method <code>BOPAlgo_PaveFiller::CheckPlanes</code> has been modified to update the filter contents by all vertices of the image faces.</p>
24266	<p><i>Summary:</i> Wrong result obtained by 'bopargcheck'</p> <p>The check of distance between the point on curve and the projected point has been added in method <code>IntTools_EdgeFace::IsProjectable</code>.</p>
24268	<p><i>Summary:</i> Wrong triangulation on the boundaries of the cones</p> <p>The following modifications have been introduced in function <code>GeomLib::NormEstim</code>:</p> <ul style="list-style-type: none"> ▪ check for cone singularity point has been added, ▪ isoline checks have been modified; ▪ check along U and V has been removed.





24286	<p><i>Summary:</i> Wrong result done by General Fuse algorithm</p> <p>The following improvements have been introduced in Fuse algorithm:</p> <ul style="list-style-type: none"> Method <code>BOPAlgo_PaveFiller::UpdatePaveBlock</code> has been modified to replace the vertices used in <code>PostTreatFF</code> (processing of section edges) with their images (new vertices created in <code>PostTreatFF</code>) in all pave blocks. Face classifier has been implemented instead of the tolerance value to define the state of a 2D point relative to the face in method <code>IntTools_Fcl ass2d::Perform</code>.
24290	<p><i>Summary:</i> BOP Section produces incorrect result</p> <p>Method <code>BOPInt_ShrunkRange</code> has been modified to reduce coefficients used to define the shrunk range for an edge with a big tolerance of its vertices</p>
24303	<p><i>Summary:</i> Precision degradation for <code>Geom2dGcc_Circ2d2TanRad</code></p> <p>The function <code>PrecRoot()</code> has been implemented in <code>GccGeo_Circ2d2TanRad</code> to precise the intersection point found by Newton method.</p>
24305	<p><i>Summary:</i> Implementation of swept shell with varying width of section bounded by auxiliary spine</p> <p>New type of resulting surface: the swept surface with varying width of section bounded by auxiliary spine has been implemented in <code>BRepOffsetAPI_MakePipeShell</code> algorithm.</p> <p>The following changes have been introduced in connection with this improvement:</p> <ul style="list-style-type: none"> New method <code>GeomFill_LocationGuide::ComputeAutomaticLaw</code> computes the set of parameter and width pairs taken from the spine and the guide. <code>BRepFill_PipeShell::Set</code> sets the flag <code><myIsAutomaticLaw></code> to True if the argument <code><KeepContact></code> is equal to <code><ContactOnBorder></code>; <code>BRepFill_PipeShell::Add</code> now adds the profile with automatically computed evolution law that interpolates a set of parameter and width pairs taken from the spine and the guide (only if <code>trihedron=GuideTrihedronAC</code> and option <code>KeepContact=ContactOnBorder</code>); Intersection of guide and surface of revolution has been replaced in <code>GeomFill_LocationGuide::SetRotation</code> by extrema between guide and surface of revolution, which is a less rigid condition. Methods <code>GeomFill_GuideTrihedronAC::D0</code> (or <code>D1</code> or <code>D2</code>) now sets the value of <code><myCurPointOnGuide></code> that is used in calculation of current width of section. New method <code>GeomFill_TrihedronWithGuide::CurrentPointOnGuide()</code> returns the current point on guide found by methods <code>D0</code>, <code>D1</code> or <code>D2</code>.
24313	<p><i>Summary:</i> <code>BRepAlgoAPI_Section</code> and <code>IntTools_FaceFace</code> are not written to handle <code>Geom_SurfaceOfLinearExtrusion</code></p> <p>The algorithm of face-to-face intersection has been corrected in class <code>IntTools_FaceFace</code>.</p>



24327	<p><i>Summary:</i> Wrong result obtained by Extrema Curve/Curve</p> <p>The number of nodes in <code>Extrema_CurveCache::Extrema_CurveCache</code> method has been increased.</p>
24328	<p><i>Summary:</i> Revolution of a wire generates two interfered faces</p> <p>Processing of cones with collinear axes has been improved in method <code>IntPatch_Intersection::Perform</code>.</p>
24359	<p><i>Summary:</i> Critical error on <code>BRepAlgoAPI_Fuse</code></p> <p>The projection algorithm <code>ProjLib_ComputeApprox</code> has been corrected to avoid critical error on <code>BRepAlgoAPI_Fuse</code>.</p>
24375	<p><i>Summary:</i> Exception in <code>IntAna</code> on touching circles</p> <p>The algorithm checking intersection of two circles has been improved in method <code>IntAna2d_AnaIntersection::Perform</code>.</p>
24384	<p><i>Summary:</i> Wrong result obtained by Fuse</p> <ul style="list-style-type: none"> ▪ Function <code>fsqrt</code> has been replaced by the standard <code>sqrt</code> in method <code>BOPTools_AlgoTools::MakeVertex</code>; ▪ Parametric deltas for computations of 2D-angles have been changed for Bezier, B-Spline 2D-curves in class <code>BOPAlgo_WireSplitter</code>.
24390	<p><i>Summary:</i> Sewing produces the result with huge tolerance</p> <p>The method <code>BRepBuilderAPI_Sewing::SameParameterEdge</code> has been corrected to avoid too big values for maximal edge tolerance.</p>
24400	<p><i>Summary:</i> Wrong result obtained by Section</p> <p>Extension of UV-bounds for faces based on spherical surfaces has been removed from function <code>IntTools_FaceFace::CorrectSurfaceBoundaries</code>.</p>
24434	<p><i>Summary:</i> The result of Boolean FUSE operation is not correct</p> <p>The algorithm splitting a seam edge on face has been improved in class <code>BOPTools_AlgoTools3D</code>.</p>
24422	<p><i>Summary:</i> Wrong result done by <code>FaceClassifier</code> algorithm</p> <p>The algorithm finding circle-point extrema has been corrected in class <code>Extrema_ExtPElC2d</code>.</p>

Visualization

23474	<p><i>Summary:</i> Changing material/color/transparency of a shaded AIS_Shape requires 'Redisplay'</p> <p>The algorithm of shaded presentation has been corrected in StdPrs_ShadedShape to properly store the material and some other properties.</p>
23486	<p><i>Summary:</i> Remove obsolete image manipulation classes</p> <p>Obsolete and unused image manipulation classes and namely the entire AlienImage package and the following classes from Image package: AveragePixelInterpolation, BalancedPixelInterpolation, BilinearPixelInterpolation, ColorImage, Convertor, GImage, GpixelField, Image, PixelInterpolation, PlanarPixelInterpolation and PseudoColorImage have been removed from OCCT.</p> <p>The functionality of removed classes has been superseded by Image_PixMap and Image_AlienPixMap classes introduced in previous OCCT releases.</p>
23501	<p><i>Summary:</i> Redundant triangulation in AIS_Shape.cxx</p> <p>AIS_Shape has been modified to clean the triangulation only in two cases:</p> <ul style="list-style-type: none"> ▪ If AIS_Shape has OwnDeviationAngle and the values newangle and prevangle are different, or ▪ If AIS_Shape has OwnDeviationCoefficient and the values newcoeff and prevcoeff are different. <p>The same behavior has been implemented in classes XCAFPrs_AISObject and AIS_TexturedShape.</p>
23654 23890 23940 23941	<p><i>Summary:</i> Problem with displaying vertices in OCC view after closing all OCC views and opening new one</p> <p>Logics and arguments of methods AIS_InteractiveContext::Erase() and AIS_InteractiveContext::EraseAll() have been changed.</p> <p>Now these methods do not remove resources from Grapgi_c3d_Structure; they simply change the visibility flag in it. This allows properly computing the erased objects again after the view is closed. Obsolete method AIS_InteractiveContext::EraseMode() has been removed.</p> <p>Additionally, Collector and all corresponding methods have been removed from AIS_InteractiveContext.</p>
23813 24392	<p><i>Summary:</i> OpenGL_Texture ignores stride image property</p> <p>New class OpenGL_UnpackAlignmentSentry has been introduced to reset unpacking alignment settings from top-to-bottom to bottom-to-top, which provides correct texture loading.</p>
23889	<p><i>Summary:</i> Assignment of function parameter has no effect outside the function</p> <p>Method Voxel_RoctBools::SetZeroSplitData() has been modified to accept a pointer address to delete data.</p>

23894	<p><i>Summary:</i> <code>Voxel_BooleanOperation(Cut)</code> gives incorrect results</p> <p>New method <code>Voxel_FastConverter::ConvertUsingSAT</code> converts a shape into a voxel presentation using separating axis theorem to compute triangle-box intersection. The intersection result allows deciding whether to set the voxel.</p>
23915	<p><i>Summary:</i> Redundant header inclusion</p> <p>Redundant <code>WNT_Window/Xw_Window</code> header inclusion has been removed from classes <code>V3d_PositionLight</code> and <code>V3d_SpotLight</code>.</p>
23917	<p><i>Summary:</i> Primitive Array with reserved but unused Edges array drawn corrupted from VBO</p> <p>The method <code>OpenGL_PrimitiveArray::BuildVBO</code> has been corrected to avoid getting a broken presentation.</p>
23959 24120	<p><i>Summary:</i> Get rid of generic classes in Visualization</p> <p>The following modifications have been introduced in <code>Prs3d</code> package to revise generic CDL classes and turn them into non-CDL C++ classes:</p> <ul style="list-style-type: none"> ▪ Generic classes <code>Prs3d_WFShape</code> and <code>Prs3d_Point</code> have been replaced with template classes; ▪ Classes <code>WFDeflectionRestrictedFace</code>, <code>CurvePresentation</code>, <code>Vector</code>, <code>Line</code>, <code>SectionShapeTool</code>, <code>PointTool</code>, <code>LineTool</code>, <code>HLRShapeTool</code>, <code>VectorTool</code>, <code>DatumTool</code> and <code>RestrictionTool</code> from <code>Prs3d</code> package have been deleted as empty and unused; ▪ Classes <code>DsgPrs_DatumPrs</code>, <code>StdPrs_WFRestrictedFace</code> and <code>StdPrs_HLRShape</code> have been replaced with simple classes as they implement generic classes from <code>Prs3d</code>.
23965	<p><i>Summary:</i> Unnecessary copies of <code>TopoDS_Face</code> in <code>Voxel_FastConverter</code> when checking triangulation</p> <p>A reference of <code>TopoDS_Face</code> is now used instead of a copy in method <code>Voxel_FastConverter::Init()</code>. Additionally it is checked if the obtained triangulation is not Null before continuing.</p>
23966	<p><i>Summary:</i> <code>Voxel_FastConverter</code> performs unnecessary triangulation.</p> <p>Method <code>Voxel_FastConverter::Init()</code> has been modified to avoid creating new triangulation if the deflection value of the existing triangulation is not greater than the specified one.</p>
24010	<p><i>Summary:</i> Getting the origin point of a voxel</p> <p>New method <code>Voxel_DS::GetOrigin</code> returns the origin point of a voxel with coordinates (<code>ix</code>, <code>iy</code>, <code>iz</code>).</p>
24013	<p><i>Summary:</i> <code>Voxel_FastConverter</code> is able to use existing triangulation</p> <p>The possibility to use existing triangulation has been implemented in <code>Voxel_FastConverter</code>.</p>

24018	<p><i>Summary:</i> Avoiding unnecessary iterations when using multiple threads</p> <p>Unnecessary iterations when using multiple threads are now avoided in method <code>Voxel_FastConverter::Convert</code>.</p>
24019	<p><i>Summary:</i> Filling problem</p> <p>New method <code>Voxel_FastConverter::FillInVolume</code> fills in a volume of the shape by a value. It uses the topological information from the provided shape to judge whether points are inside the shape or not (only when processing vertical faces). This method allows processing of vertical faces and concave shapes.</p>
24031	<p><i>Summary:</i> Depth buffer writes not enabled after rendering a transparent object</p> <p>The handling of depth buffer writes has become independent from the reflection mode flags to provide a correct depth buffer operation after rendering transparent object(s) in method <code>OpenGL_Workspace::UpdateMaterial</code>.</p>
24050	<p><i>Summary:</i> Problems in multithreaded scenario</p> <p>The methods <code>Convert</code> and <code>ConvertUsingSAT</code> from <code>Voxel_FastConverter</code> have been corrected to improve computation of triangle spans.</p>
24070 24189 24270	<p><i>Summary:</i> OpenGL capped object-level clipping planes</p> <p>Object-level clipping and capping algorithm has been introduced in OCCT. See the New Features section for its description.</p> <p>The following modifications have been introduced in connection with this improvement:</p> <ul style="list-style-type: none"> ▪ New <code>Graphi c3d_ClipPlane</code> class defines plane equation, clipping properties and visual attributes for capping. ▪ Low-level API methods <code>SetClipPlanes()</code> and <code>GetClipPlanes()</code> have been added to class <code>Graphi c3d_Structure</code> to define clipping per structure. ▪ Methods <code>PrsMgr_Presentabl eObject::AddClipPlane()</code>, <code>RemoveClipPlane()</code>, <code>SetClipPlanes()</code> and <code>GetClipPlanes()</code> should be used to manage per-object clipping. ▪ Old methods for registering <code>V3d_Plane</code> instances: <code>AddPlane()</code>, <code>DelPlane()</code>, <code>Ini tDefi nedPlanes()</code>, <code>MoreDefi nedPlanes()</code>, <code>NextDefi nedPlane()</code> and <code>Defi nedPlane()</code> have been removed. ▪ Methods <code>AddClipPlane()</code>, <code>RemoveClipPlane()</code>, <code>SetClipPlanes()</code> and <code>GetClipPlanes()</code> have been added in <code>V3d_Vi ew</code> package to define global clipping according to <code>Graphi c3d_ClipPlane</code> approach. ▪ <code>V3d_Plane</code> class has been ported to <code>Graphi c3d_ClipPlane</code> approach – it provides graphical representation of clip plane and interface to change its <code>Graphi c3d_ClipPlane</code> definition. ▪ The argument list of methods for picking detection has been changed in methods <code>SelectBasi cs_Sensi ti veEntity::Matches()</code> and <code>Select3D_Sensi ti veEntity::Matches()</code>. ▪ The class <code>SelectBasi cs_Pi ckArgs</code> has been added to provide extended set of arguments for <code>Matches</code> method. ▪ Methods <code>GetEyeLine()</code>, <code>Depth()</code>, <code>SetLastPrj()</code> and <code>SetLastDepth()</code> have been removed from class <code>Select3D_Sensi ti veEntity</code>. ▪ Methods <code>DepthMi n()</code>, <code>DepthMax()</code> and <code>DepthMi nMax()</code> have been removed from class <code>Select3D_Projector</code>. ▪ Abstract method <code>Select3D_Sensi ti veEntity::ComputeDepth()</code> has been removed, the depth checks now should be done by picking <code>Matches()</code>;

<p>24070 24189 24270</p>	<ul style="list-style-type: none"> ▪ Virtual callbacks for application-level implementations: <code>PickingLine()</code>, <code>DepthClipping()</code> and <code>HasDepthClipping()</code> have been added to <code>SelectMgr_ViewerSelector</code> to provide arguments for picking detection. ▪ Methods <code>PickingLine()</code>, <code>DepthClipping()</code>, <code>HasDepthClipping()</code> have been implemented in class <code>StdSelect_ViewerSelector3d</code>. ▪ Classes <code>OpenGL_CappingAlgo</code>, <code>OpenGL_CappingPlaneResource</code>, <code>OpenGL_ClippingState</code> and <code>OpenGL_RenderFilter</code> have been added to provide support for capping rendering algorithm and per-structure clipping. ▪ DRAW command <code>vclipplane</code> has been redesigned for new approach. ▪ DRAW command <code>vsettexturemode</code>, which enables texturing in the view, has been added. ▪ MFC samples have been ported on the new clipping planes implementation.
<p>24113</p>	<p><i>Summary:</i> Provide missing <code>OpenGL_VertexBuffer::SubData()</code> specializations</p> <p>New functions <code>SubData</code> providing <code>OpenGL_VertexBuffer::SubData()</code> specializations have been added in class <code>OpenGL_VertexBuffer</code>.</p>
<p>24123 24230 24233 24245 24256</p>	<p><i>Summary:</i> Create debug OpenGL context when requested (<code>GL_ARB_debug_output</code>)</p> <p>Debug OpenGL context provides more feedback concerning low-level visualization issues. By default, the new mode is turned ON for debug build and turned OFF for release build.</p> <p>The following modifications have been introduced in connection with this improvement:</p> <ul style="list-style-type: none"> ▪ New class <code>OpenGL_Caps</code> has been added to provide unified access for advanced graphic driver options, including VBO usage, creation of debug GL context and some others. ▪ New command <code>vgl debug</code> has been added to request debug OpenGL context. ▪ <code>OpenGL_Window</code> class now creates debug OpenGL context when requested (currently implemented only for Windows). ▪ In addition memory leaks have been eliminated in <code>OpenGL_Context</code> class: members <code>arbTBO</code> and <code>arbIns</code> are removed. ▪ Overcomplicated <code>find_pixel_format()</code> function has been removed from <code>OpenGL_Window</code> class; ▪ The option to choose software OpenGL implementation (MS or Apple) has been implemented in <code>OpenGL_Caps</code> class.
<p>24128</p>	<p><i>Summary:</i> Remove debug environment switch <code>DEBUG_TRIANGLES</code></p> <p>Unused debug environment switch <code>DEBUG_TRIANGLES</code> has been removed from class <code>Prs3d_WFShape</code>.</p>
<p>24130 24353 24459</p>	<p><i>Summary:</i> Implementation of ray tracing visualization core</p> <p>Realistic rendering using ray tracing method can now be activated for shaded objects in 3D viewer. All ray-tracing computations are performed on the GPU using OpenCL framework, allowing real-time rendering performance. Real-time ray-tracing is possible using high-performance GPUs with support of OpenCL 1.1 and higher</p> <p>Currently ray-tracing visualization core supports sharp shadows, specular reflections, transparency and adaptive anti-aliasing. However, the basis for all ray-tracing algorithms is versatile and new ray-tracing features (such as ambient occlusion) can be added easily.</p>



<p>24130 24353 24459</p>	<p>The following methods are used to enable/disable ray-tracing mode in a specific View:</p> <ul style="list-style-type: none"> • <code>V3d_View::SetRaytracingMode()</code> – enables ray-tracing rendering mode. • <code>V3d_View::SetRasterizationMode()</code> – enables OpenGL (rasterization) rendering mode. <p>To enable/disable specific ray-tracing features, the following methods are used:</p> <ul style="list-style-type: none"> • <code>V3d_View::EnableRaytracedShadows()</code> and <code>V3d_View::DisableRaytracedShadows()</code> – enable/disable ray-traced sharp shadows. • <code>V3d_View::EnableRaytracedReflections()</code> and <code>V3d_View::DisableRaytracedReflections()</code> – enable/disable ray-traced specular reflections. • <code>V3d_View::EnableRaytracedAntialiasing()</code> and <code>V3d_View::DisableRaytracedAntialiasing()</code> enable/disable ray-traced adaptive anti-aliasing. <p>Qt IESample provides an example of OCCT-based application with support of ray-tracing.</p>
<p>24131 24148 24158 24455</p>	<p><i>Summary:</i> Redesign of GPU memory management for markers presentation</p> <p>Texture-based point sprites are now used by default for markers on modern hardware. This improvement replaces previously used inefficient bitmaps and significantly improves rendering performance of big marker arrays.</p> <p>Additionally, marker definition now supports full-color images.</p> <p>The issues with GPU memory management of marker resources (broken sharing between multiple displays, memory leaks and broken usage after 3D view destruction / re-creation) have also been fixed.</p>
<p>24133 24389</p>	<p><i>Summary:</i> Implementation of dimensions</p> <p>The look and construction ergonomics for AIS dimension presentation classes have been improved.</p> <p>Extra construction arguments have been added to <code>AIS_LengthDimension</code>, <code>AIS_AngleDimension</code> and <code>AIS_DiameterDimension</code>:</p> <ul style="list-style-type: none"> • Length dimension can be built on edge and between two points. • Diameter (radius) dimension can be built for a shape containing circle geometry. • Angle dimension can be built between two edges or by three points. <p>Old construction arguments have been revised: there is no more necessity to pass text string with dimension value. The dimensions features are external and extension lines, 2D & 3D text, 2D & 3D arrows, text positioning and alignment along the extension base line, programmatic computation of dimension value and unit conversion. The common properties are encapsulated within <code>AIS_Dimension</code> base class.</p> <p>Dimension aspects have been revised to group separate <code>Prs3d_AngleAspect</code> and <code>Prs3d_LengthAspect</code> aspects under <code>Prs3d_DimensionAspect</code>.</p> <p>The enumeration <code>AIS_DimensionSelectionMode</code> has been implemented to enumerate allowed selection modes for dimension presentations. It replaces the obsolete <code>AIS_DimensionDisplayMode</code> enumeration.</p> <p><code>OpenGL_Window</code> and <code>Xw_Window</code> classes now request support of stencil buffer when configuring pixel format.</p> <p>DRAWEXE “vdim” command has been added to test improved dimension presentations.</p>





<p>24181</p>	<p><i>Summary:</i> Text to Brep functionality</p> <p>New class <code>Font_BrepFont</code> has been introduced for conversion of font glyph in vector format into Brep representation (<code>TopoDS_Shape</code>). A single instance of this class initializes a single font for sequential rendering of glyphs with implicit caching of already rendered glyphs.</p> <p>The method <code>RenderText()</code> creates Brep representation of the specified text at a given location in 3D space. The options are provided to create glyphs as faces or wires (outline) and to use C0 or C1 curves.</p>
<p>24188</p>	<p><i>Summary:</i> The number of light sources in Qt samples</p> <p>Method <code>V3d_Viewer::SetDefaultLights()</code> has been fixed to avoid adding new light sources at each call.</p>
<p>24192 24241 24352</p>	<p><i>Summary:</i> Support of shaders in OCCT visualization toolkit</p> <p>Support of GLSL shaders has been implemented OCCT visualization core. Shaders can provide various rendering effects, such as per-pixel lighting, custom shading models, bump mapping, procedural texturing and others, on graphics hardware with a high degree of flexibility. Currently OCCT supports only vertex and fragment GLSL shaders, but this can be extended in the future.</p> <p>The following Draw commands allow working with shaders:</p> <ul style="list-style-type: none"> ▪ <code>vshaderprog <ShapeName> <Path to Vertex Shader> <Path to Fragment Shader></code> allows setting the custom vertex and fragment GLSL shaders; ▪ <code>vshaderprog <ShapeName> off</code> allows detaching shader program from a specific shape; <p>Note, that OpenGL 2.0+ GPU is required to execute shader programs.</p> <p>To enable a custom shader for a specific <code>AISShape</code> in the application, the following API functions are used:</p> <ul style="list-style-type: none"> ▪ <code>Handle(Graphic3d_ShaderProgram) aProgram = new Graphic3d_ShaderProgram()</code> allows creating shader program; ▪ <code>aProgram->AttachShader (Graphic3d_ShaderObject::CreateFromFile(Graphic3d_TOS_VERTEX, "<Path to VS>"))</code> allows attaching a vertex shader; ▪ <code>aProgram->AttachShader (Graphic3d_ShaderObject::CreateFromFile(Graphic3d_TOS_FRAGMENT, "<Path to FS>"))</code> allows attaching a fragment shader; ▪ <code>aProgram->PushVariable ("MyColor", Graphic3d_Vec3(0. f, 1. f, 0. f))</code> allows setting values for custom uniform variables (if exist) ▪ <code>AISShape->Attributes() ->ShadingAspect() ->Aspect() ->SetShaderProgram (aProgram)</code> allows setting aspect property for a specific <code>AISShape</code>.
<p>24198</p>	<p><i>Summary:</i> Remove unused stuff in WNT package</p> <p>Obsolete win32 and Windows 95 specific files have been removed from WNT package.</p>
<p>24224 24229 24253</p>	<p><i>Summary:</i> Suspicious logics in changing clipping planes at <code>OpenGL_Structure</code></p> <p><code>OpenGL_ClipPlaneState</code> and <code>OpenGL_Structure</code> have been fixed to show correctly the clip planes shared between the view and at least two structures.</p> <p><code>OpenGL_FrameBuffer::Init</code> has been modified to use stencil buffer for image dump functionality in 3D viewer.</p>





24225	<p><i>Summary:</i> Wrong ID is used for attachment of depth <code>RenderBuffer</code> to FBO</p> <p>The method <code>OpenGL_FrameBuffer::Init</code> has been corrected to use correct IDs for attachment of depth <code>RenderBuffer</code> to FBO. Previously this issue might result in a broken 3D viewer image dump.</p>
24228 24297	<p><i>Summary:</i> TKOpenGL – destroy GL context at view close</p> <p>The following improvements have been introduced to avoid keeping context infinitely in an <code>OpenGL_Display</code> instance.</p> <ul style="list-style-type: none"> ▪ <code>OpenGL_Display</code> release GL resources correctly on closing views; ▪ <code>OpenGL_AspectFace</code>, <code>OpenGL_AspectText</code>, <code>OpenGL_AspectLine</code>, and <code>OpenGL_AspectMarker</code> have been modified to initialize OpenGL resources on demand, when the context is available; ▪ <code>Graphic3d_TextureRoot</code> uses <code>const</code> modifier for <code>GetId</code> method to avoid asynchronous resource state at OpenGL; ▪ Calls of OpenGL functions are avoided if no active GL context exists.
24250 24309 24344 24345 24348	<p><i>Summary:</i> TKOpenGL – per-pixel lighting using GLSL program (Phong shading)</p> <p>OpenGL/GLSL vertex and fragment shaders have been implemented to enable per-pixel lighting (Phong shading).</p> <p>Phong shading (or normal-interpolation shading) overcomes some of the disadvantages of Gouraud shading implemented in the OpenGL rendering pipeline. This technique performs per-fragment light calculations within GLSL program. At each fragment the interpolated surface normal is used to evaluate light intensity. Interpolation of normal allows highlights smaller than a polygon.</p> <p>The normal-interpolation shaders are implemented as a part of OCCT visualization core (see <code>Graphic3d_ShaderProgram::ShaderName</code> enumeration).</p> <p>Lighting implementation has been improved to work with Phong shaders correctly. Per-fragment lighting computations can be enabled using the DRAW command <code>vshaderprog [name] phong</code>.</p> <p>To enable normal-interpolation shading in the application, see the source code of <code>VshaderProg</code> function (in <code>TKViewerTest</code> toolkit).</p> <p>New command <code>vdefaults</code> allows controlling automatic triangulation parameters (deflection value and absolute/relative mode).</p>
24267	<p><i>Summary:</i> Exception in <code>Visual3d_ViewManager::Redraw()</code> when color scale is displayed</p> <p>Obsolete <code>Init()</code>, <code>DefaultOrthographicView()</code> and <code>DefaultPerspectiveView()</code> methods have been removed from <code>V3d_Viewer</code> class.</p>
24276	<p><i>Summary:</i> Memory leak due to a static variable</p> <p><code>AIS_Selection</code> and <code>AIS_InteractiveContext</code> have been modified to avoid memory leaks.</p>



24282	<p><i>Summary:</i> Shaded presentation is not computed due to exception in <code>StdPrs_ToolShadedShape::Normal()</code></p> <p><code>gp_Dir</code> initialization with zero modulus has been fixed.</p>
24288 24374	<p><i>Summary:</i> Provide flipping text for <code>AIS_Dimensions</code></p> <p>New <code>OpenGL_Flipper</code> class provides flipping for subsequent elements in current graphic group. It provides proper display of objects according to the current view orientation by applying proper transformations to the objects.</p> <p><code>SetFlippingOptions()</code> methods have been added in <code>Graphic3d_Group</code>, <code>Graphic3d_GraphicDriver</code> and <code>OpenGL_GraphicDriver</code> classes to provide interfaces for enabling and disabling of flipping options to the graphic group.</p> <p><code>Graphic3d_Mat4</code> and <code>Graphic3d_Mat4d</code> definitions have been added for use with <code>TKOpenGL</code> matrix types.</p>
24293	<p><i>Summary:</i> Dimension flyout lines don't belong to the dimension sensitive entity.</p> <p>Function <code>computeFlyoutSelection()</code> has been implemented in <code>AIS_Dimension</code> class to add flyout sensitive in the default selection mode and in <code>AIS_AngleDimension</code> class to provide selection of specific flyout lines.</p> <p>Correspondingly flyout field has been moved to <code>AIS_Dimension</code> class as a property common for all dimensions.</p>
24294	<p><i>Summary:</i> Access violation in <code>AIS_InteractiveContext::MoveTo()</code></p> <p>Access violation has been fixed in method <code>AIS_InteractiveContext::MoveTo()</code>.</p>
24310 24322	<p><i>Summary:</i> GLSL compatibility issues</p> <p>Light sources definition has been improved:</p> <ul style="list-style-type: none"> ▪ duplicate enumeration <code>TLightType</code> has been removed (the same as <code>Visual3d_TypeOfLightSource</code>); ▪ unused fields have been removed from <code>Graphic3d_CLight</code>; ▪ <code>Graphic3d_CLight</code> definition is reused in <code>OpenGL_Light</code> <p>The following changes have been introduced in <code>Phong</code> GLSL program:</p> <ul style="list-style-type: none"> ▪ Cumulative ambient light intensity has been excluded from a limited list of lights; ▪ Compatibility issues with old OpenGL implementations have been fixed; ▪ Arrays of structures (light sources, materials, clipping planes) have been replaced with arrays of primitive types. ▪ Handling of multiple ambient light sources has become consistent between pipeline rendering and GLSL program / Ray Tracing. <p>New Draw command <code>vlight</code> allows altering light sources definition.</p>
24311	<p><i>Summary:</i> <code>TKOpenGL</code> – disappearing 2D dimension text</p> <p>Stencil buffer filling has been corrected in class <code>OpenGL_Text</code>.</p>



24321	<p><i>Summary:</i> Use <code>List</code> collection instead of <code>NCollection_Set</code> for collection of <code>Handle(Graphic3d_Cl i pPlane)</code></p> <p><code>NCollection_Set (Graphic3d_SetOfHCl i pPlane)</code> has been replaced by ordered <code>NCollection_Sequence</code> type (<code>Graphic3d_SequenceOfHCl i pPlane</code>) for management of collections of planes to guarantee that the objects will be returned in the order in which they were added.</p>
24315	<p><i>Summary:</i> Use delayed release of resources in <code>OpenGL</code> aspects</p> <p>Delayed release has been implemented for face texture resource of <code>OpenGL_AspectFace</code> and point sprite texture of <code>OpenGL_AspectMarker</code> instead of instant release.</p>
24326	<p><i>Summary:</i> Get rid of confusing extension line</p> <p>The following improvements have been introduced in <code>AIS_Dimensi on</code> class:</p> <ul style="list-style-type: none"> ▪ Duplicate property <code>TextOffset</code> has been removed; ▪ <code>ExtensionSi ze</code> has been redefined to affect the length of Left/Right extensions and moved to <code>Prs3d_Dimensi onAspect</code>; ▪ The confusing line tip is not drawn anymore; ▪ Text alignment properties have been replaced with items from <code>Prs3d_Dimensi onTextHori zontal Posi ti on</code> and <code>Prs3d_Dimensi onTextVerti cal Posi ti on</code> enumerations. <p>Automatic label positioning and arrow orientation features now can be controlled an appropriate value from the related enumerations:</p> <ul style="list-style-type: none"> ▪ Specifying <code>Prs3d_DAO_Fit</code> as <code>ArrowOri entati on</code> automatically determines orientation of the arrows depending on the size of text label; ▪ Specifying <code>Prs3d_DTHP_Fit</code> as <code>TextHori zontal Posi ti on</code> automatically determines label position arrows depending on the text label.
24350	<p><i>Summary:</i> Non-physical materials are ignored by GLSL program manager</p> <p>Shader Manager and Ray Tracing have been improved to take non-physical materials into account.</p>
24386	<p><i>Summary:</i> High-level API to specify font by user-defined path for <code>AIS (Prs3d)</code> aspects</p> <p>New method <code>Font_FontMgr::RegisterFont()</code> has been implemented to register custom fonts.</p> <p>Draw command <code>vfont</code> gives access to OCCT font manager:</p> <ul style="list-style-type: none"> ▪ To print the list of all available fonts, type <code>vfont</code> without arguments; ▪ To add a new font, type <code>font add <file> <font_name></code>, where <code><file></code> is the path to a <code>.ttf</code> format file, for example, <code>c:/Windows/Fonts/Dej aVuSeri fCondensed. ttf</code>, and <code><font_name></code> is a custom name assigned to this font, for example, <code>fnt1</code>; ▪ To search for a specific font, type <code>vfont fi nd <font_name></code>.





24420	<p><i>Summary:</i> Switch the type of sensitivity in AIS_Circle and AIS_Plane</p> <p>The following methods have been implemented to switch the type of sensitivity in AIS_Circle and AIS_Plane:</p> <ul style="list-style-type: none"> ▪ AIS_Plane: : TypeOfSensitivity - returns the type of sensitivity for the plane; ▪ AIS_Plane: : SetTypeOfSensitivity - sets the type of sensitivity for the plane. ▪ AIS_Circle: : IsFilledCircleSens - returns the type of sensitivity for the circle ▪ AIS_Circle: : SetFilledCircleSens - sets the type of sensitivity for the circle. If theIsFilledCircleSens is Standard_True, the whole circle will be detectable, otherwise only the circle boundary will be detectable.
24425	<p><i>Summary:</i> Improve ergonomics of new dimension presentations</p> <p>The following user-interface methods and rules simplifications have been introduced to AIS_Dimension classes to improve the ergonomics:</p> <ul style="list-style-type: none"> ▪ It has become possible to get the value, drawing plane and measured geometry properties before the presentation is displayed. The corresponding methods have been added to AIS_AngleDimension, AIS_LengthDimension, AIS_DiameterDimension and AIS_RadiusDimensions classes. ▪ Methods SetMeasuredGeometry have been added to the same classes to allow changing the measured geometry of already constructed presentations. ▪ The method AIS_Dimension: : IsValid() has been added to check whether the geometrical parameters of dimension presentations are valid (measured geometry, plane). ▪ The unit definition for conversion is now specified by Prs3d_DimensionUnits class. The instance is provided as presentation attribute by Prs3d_Drawer. The drawer mechanism allows specifying default application units and overriding them for particular presentations. ▪ The format string for formatting units label value has been added to Prs3d_DimensionAspect class. ▪ The setter methods have been revised to invoke SetToUpdate calls each time the property is changed to refresh it on the next update.
24450	<p><i>Summary:</i> TKOpenGL - Fixed pipeline materials update optimization does not distinguish front/back faces</p> <p>OpenGL material definition has been improved: the values for various types of light (Ambient, Diffuse, Specular and Emission) can be properly define for front and back faces separately.</p>
24452	<p><i>Summary:</i> TKOpenGL - Memory leak in OpenGL_ShaderManager</p> <p>aLightTypeArray is now properly removed from memory in method OpenGL_ShaderManager: : PushLightSourceState.</p>



Data Exchange

22535	<p><i>Summary:</i> Cut sphere: invalid shape after reading from STEP</p> <p>Method <code>ShapeFix_Wire::FixShifted()</code> has been fixed to produce correct parametric representation.</p>
23746	<p><i>Summary:</i> IGES wheel model fails to load when OCCT unit is meters</p> <p>The following improvements have been introduced into the algorithm of saving to IGES:</p> <ul style="list-style-type: none"> ▪ Check of outer boundary type has been added in method <code>IGESGeom_ToolTrimmedSurface::WriteOwnParams</code>. ▪ Too small distances between loops of edges are now properly checked when the edges are reordered. ▪ Too small B-spline segments are now checked during reading and writing IGES. ▪ It is checked if vertices are the same not only in 3D, but also in 2D during wire splitting.
23771	<p><i>Summary:</i> Writing offset-based surfaces of revolution to IGES</p> <p>New parameter <code>write.iges.offset.mode</code> added in class <code>GeomToIGES_GeomCurve</code> allows writing offset curves in form of b-splines.</p>
23846 24176	<p><i>Summary:</i> Array out-of-bound error in Vrml loader</p> <p>The code which assigns normals to triangulation nodes has been corrected to use correct index of the node and avoid array out-of-bound errors in method <code>VrmlData_IndexedFaceSet::Tshape</code>.</p>
23921	<p><i>Summary:</i> IGES reader cannot map subshapes colors inside nested assemblies</p> <p>The function <code>IGESCAFControl_Reader::AddCompositeShape</code> has been modified to avoid creation of redundant labels in XCAF for any shapes belonging to a compound even if colors are not defined for them.</p>
23938	<p><i>Summary:</i> Crash at <code>StepToTopoDS_TranslateEdgeLoop::Init()</code> and <code>StepToTopoDS_TranslateEdge::Init()</code></p> <p>Check of curve type has been added in method <code>StepToTopoDS_TranslateEdgeLoop::Init()</code> to avoid crash.</p>
23996	<p><i>Summary:</i> Width 50 given in format string is larger than destination buffer 'value[50]'</p> <p>The size of the destination buffer has been increased in <code>Interface_Static</code> class to avoid possible overflow.</p>
24007	<p><i>Summary:</i> Sporadic crashes when working with XDE documents</p> <p>The function <code>GetRefID</code> has been added in <code>XCAFDoc_DocumentTool</code> to return the reference ID and to find a tree node attribute at the root label.</p>



24077	<p><i>Summary:</i> Build errors on master branch on Mac OS X</p> <p>Code for construction of faces has been corrected in <code>StepToTopoDS_Builder</code> and <code>IGESToBRep_TopoSurface</code> classes.</p>
24135	<p><i>Summary:</i> Result of reading step file is invalid.</p> <p>STEP to Geom conversion algorithm has been changed to limit multiplicity of boundary knots to <code>Degree + 1</code> (if it is higher due to invalid data contained in STEP file).</p>
24206	<p><i>Summary:</i> Exception is raised in the STEP loopback tests</p> <p>Classes <code>ShapeFix_EdgeProjAux</code> and <code>ShapeAnalysis_Curve</code> have been modified to avoid potentially uninitialized variables in Shape Healing.</p>
24279	<p><i>Summary:</i> Correction in <code>STEPCAFControl_Reader::Transfer</code></p> <p>A parameter has been corrected in <code>STEPCAFControl_Reader::Transfer</code> method.</p>
24283	<p><i>Summary:</i> STEP read fails to read unbound <code>advanced_face</code></p> <p>Method <code>RWStepShape_RWAdvancedFace::ReadStep</code> has been improved to avoid exception if a sphere is written without bounds.</p>
24383	<p><i>Summary:</i> Access violation during STEP file import</p> <p>The method <code>StepToGeom_MakeLine::Convert</code> has been protected against null vector magnitude.</p>
24448	<p><i>Summary:</i> Possible copy-paste error in <code>IGESGeom_SplineCurve.cxx</code></p> <p>A typo has been fixed in method <code>IGESGeom_SplineCurve::IGESGeom_SplineCurve()</code>.</p>

Draw

23570	<p><i>Summary:</i> Command <code>param</code> in DRAW outputs non-readable symbols</p> <p>The output has been corrected in <code>MoniTool_TypedValue</code> class.</p>
23670 24136 24295	<p><i>Summary:</i> Support for multiple 3D views</p> <p>DRAW has been improved to support creation of several 3D views and removing a 3D view programmatically.</p> <p>The following commands for management of several views are available:</p> <ul style="list-style-type: none"> ▪ <code>vnit</code> – creates a new 3D view instance each time it is executed. It accepts an optional argument telling if the existing <code>V3d_Viewer/AIS_InteractiveContext</code> should be reused for the new view. This command returns the view identifier. ▪ <code>vclose</code> – closes a given 3D view instance or the active one. ▪ <code>vactivate</code> – activates the given 3D view instance. The active view is the one to which commands such as <code>vnit</code> will be applied.



23904	<p><i>Summary:</i> Multiple warnings on Windows x86_64 target</p> <p>The warnings concerning <code>WinAPI</code> usage have been fixed in Draw package.</p>
23906	<p><i>Summary:</i> Performance of the projection algorithm in some cases became lower</p> <p>Draw command <code>proj ponf</code> has been modified to allow changing the default parameters of <code>Extrema_ExtPS</code> algorithm.</p>
23960	<p><i>Summary:</i> Some <code>ViewerTest</code> commands lack description</p> <p>Missing descriptions that can be returned by <code>'help'</code> command have been added in the following commands from <code>ViewerTest</code> package: <code>vdisplay</code>, <code>verase</code>, <code>vdonly</code>, <code>vdisplayall</code>, <code>veraseall</code>, <code>vsetdisplaymode</code>, <code>vunsetdisplaymode</code>, <code>vdirect</code>, <code>vsetcolor</code>, <code>vunsetcolor</code>, <code>vsettransparency</code> <code>vunsettransparency</code> and <code>vstate</code>.</p>
24029	<p><i>Summary:</i> Possibility to disable Face/Face intersection</p> <p>The command <code>bopcheck</code> now uses an additional parameter that can take values from 0 to 5. It defines which intersections should be checked, for example:</p> <ul style="list-style-type: none"> ▪ <code>bopcheck shape 1</code> sets the level of check to V/E, i.e. intersections #E/E, V/F, E/F, F/F will not be checked. ▪ <code>bopcheck shape 4</code> sets the level of check to E/F, i.e. intersection F/F is disabled. ▪ <code>bopcheck shape 5</code> checks all intersections (same as <code>bopcheck shape</code>).
24061	<p><i>Summary:</i> Draw-commands for Voxels are insufficient to test the functionality</p> <p>The draw commands for voxels have been upgraded by parameters (previously they had none). New <code>VOXELBOOLDSCONVERT</code> command has been added to call the converter of a shape into voxels.</p>
24115	<p><i>Summary:</i> Draw-command <code>loadvrml</code> fails on non-current folder</p> <p>The algorithm that finds the path to VRML file has been added in the Draw command.</p>
24132	<p><i>Summary:</i> Test execution fails due to permission rights issues</p> <p>Generation of temporary files used for capturing DRAW command output on Windows has been improved to use the directory specified by <code>TMP</code> or <code>TEMP</code> environment variable (instead of the root directory, which can be write-protected).</p>
24244	<p><i>Summary:</i> Command <code>bclear</code> in DRAW does not clear all the data required for BOP re-initialization</p> <p><code>BOPTest_Objects::Clear()</code> has been extended to also clear the arguments (objects and tools) of the operation..</p>
24246	<p><i>Summary:</i> Draw command <code>stepread</code> without arguments throws an exception</p> <p>Check for the number of arguments has been added in <code>stepread</code> command.</p>

24277	<p><i>Summary:</i> DRAW command that shows coordinates of a vertex</p> <p>New command <code>getcoords</code> allows printing the co-ordinates of one or several vertices given as arguments.</p>
24316	<p><i>Summary:</i> Make building with different versions of Tcl easier</p> <p>On Windows with MSVC the Tcl library is linked using <code>#pragma</code> statement instead of project properties, to ease switch between Tcl versions.</p>
24337	<p><i>Summary:</i> Relax <code>vi ni t</code> syntax</p> <p>The command <code>vi ni t</code> now supports a simplified syntax: <code>vi ni t vi ewName</code> instead of <code>vi ni t name=vi ewName</code>.</p>
24391	<p><i>Summary:</i> Erased AIS object cannot be displayed in <code>AIS_InteractiveContext</code> after <code>AIS_InteractiveContext::Remove()</code></p> <p>The behavior of <code>vremove</code> command has been corrected.</p>
24396	<p><i>Summary:</i> <code>vsel mode</code> – disable auto loading of objects into Local Context</p> <p>The command <code>vsel mode</code> has been modified to add to the local selection only the specified objects.</p>
24451	<p><i>Summary:</i> Logical conjunction always evaluates to false</p> <p>When a variable is cast to <code>Standard_Byte</code> it is now checked to be within the range <code><0; 255></code>.</p>

Mesh

23105	<p><i>Summary:</i> Exception during Meshing / Missing triangles</p> <p>New functionality for resolving edge intersections and loops on polygon level has been implemented in <code>BrepMesh_Delaun</code> class. The algorithm does not split existing mesh links by adding new points, but tries to resolve problems by replacing a set of problematic frontier links by a single link or by dividing the source polygon into simpler ones.</p>
23946	<p><i>Summary:</i> Uninitialized variable <code>aNewEdge3</code> used</p> <p>The method <code>BrepMesh_Delaun::CreateTriangles</code> has been modified to correctly initialize <code>aNewEdge3</code> variable.</p>



Shape Healing

21317	<p><i>Summary:</i> Face cannot be fixed by Shape Healing</p> <p>The check for edges that were only shifted reverse or forward (not reordered) has been added in method <code>ShapeAnalysis_WireOrder::Perform</code>.</p>
23944	<p><i>Summary:</i> Wrong approximation results</p> <p>A typo has been fixed in <code>ShapeCustom_RestrictionParameters</code>.</p>
24008	<p><i>Summary:</i> <code>ShapeAnalysis_Surface</code> causes exception in <code>Geom_OffsetSurface</code></p> <p>The algorithm searching for point projection in <code>ShapeAnalysis_Surface::SurfaceNewton()</code> now stops when the found point is beyond the surface domain.</p>
24105	<p><i>Summary:</i> <code>ShapeFix</code> algorithm produces not correct result</p> <p><code>ShapeFix_Face</code> algorithm now does not change the status of faces, if the edges in their wires were only shifted, not reordered.</p>
24126	<p><i>Summary:</i> Crash on fixing the attached shape</p> <p>The check for an empty list has been added in function <code>ShapeFix_Face::FixSplitFace</code>.</p>
24385	<p><i>Summary:</i> Incorrect flag <code>isClosed</code> for shell</p> <p>The method <code>ShapeFix_Shell::FixFaceOrientation</code> has been modified to set flag <code>isClosed</code> correctly.</p>
24403	<p><i>Summary:</i> <code>BRepBuilderAPI_Sewing</code> should have get/set Context methods</p> <p>Methods to get and set context have been introduced in <code>BRepBuilderAPI_Sewing</code> class.</p>

WOK

23930	<p><i>Summary:</i> Location of GLX in Cmake meta-project of OCCT</p> <p>It has become possible to define the location of GLX with the variable <code>3RDPARTY_GLX_DIR</code> in Cmake build scripts.</p>
23975	<p><i>Summary:</i> Remove <code>CSF_Graphi cShr</code> environment variable</p> <p>Unused <code>CSF_Graphi cShr</code> environment variable has been removed from <code>env. bat</code> generated by WOK.</p>
24090	<p><i>Summary:</i> 32-bit debug version of DRAWEXE is linked against release CRT libraries when compiled from project files generated for VS2008</p> <p>Project file template "<code>template. vc9x</code>" has been fixed to provide correct linking.</p>
24153	<p><i>Summary:</i> OpenCL implementation</p> <p>OpenCL resource directory (<code>CSF_OpenCLDir</code>) has been added to WOK scripts. The WOK GUI has also been modified (include, lib and bin directories of the OpenCL SDK added).</p>
24160	<p><i>Summary:</i> Changes in VS Projects warning level</p> <p>Cmake, Automake and MSVC compiler scripts have been upgraded to set maximum warning level (<code>-W4</code> for MSVC, <code>-Wall</code> for GCC).</p>
24165	<p><i>Summary:</i> Eliminate compiler warnings on unused variables in files generated by WOK (Schema)</p> <p><code>CSFDBSchema_Template.edl</code> (the template for standard persistence files generated by WOK) has been corrected to avoid compiler warnings in such files.</p>
24175	<p><i>Summary:</i> Generate <code>DRAWEXE.vcxproj.user</code> to launch DRAWEXE for vc10/vc11 projects</p> <p><code>DRAWEXE.vcxproj.user</code> configuration file are now created by WOK to launch DRAWEXE from generated vc10/vc11/vc12 projects (previously only vc9 was supported).</p>
24214	<p><i>Summary:</i> <code>wgenproj</code> command started on Linux platform with <code>-target=amk</code> option generates <code>env. sh</code> file for MacOS only</p> <p>The behavior of <code>wgenproj</code> command has been corrected to generate <code>env. sh</code> files for all supported platforms.</p>
24308	<p><i>Summary:</i> Enable option <code>/fp: precise</code> explicitly in Visual Studio projects</p> <p>Option <code>/fp: precise</code> has been explicitly enabled in Visual Studio projects generated by WOK. It protects against non-safe optimization of floating-point arithmetic, which is applied by default if VS projects are converted to use Intel compilers.</p>

24314	<i>Summary:</i> Generated env.sh defines Release libraries search paths for Debug target Incorrect path definition has been fixed in draw.sh template.
24333	<i>Summary:</i> WOK environment fix WOK command wenv has been corrected to properly setup the path to UnitsAPI package in the environment.
24366	<i>Summary:</i> SPRODUCTS_PATH ignores platform SPRODUCTS_PATH has been corrected to take the platform into account.
24369	<i>Summary:</i> Specify Tcl/Tk as frameworks rather than libraries on OS X -ltcl86 and -ltk86, which are not pre-installed on OS X, are now specified as frameworks on this platform.
24378	<i>Summary:</i> Empty aTclHPATH variable when tcl headers are located at /usr/include/tcl8.6/ Script wok_deps.tcl has been modified to correctly empty aTclHPATH variable.

Build

23947 24057 24058 24059 24085 24087 24095 24096 24129 24186	<p><i>Summary:</i> Eliminate MSVC++ 2008 compiler warnings with warning level 4</p> <p>The OCCT code has been revised to get rid of existing MSVC++ 2008 compiler warnings of the following types:</p> <ul style="list-style-type: none"> ▪ C4063: case 'identifier' is not a valid value for switch of enum 'enumeration'; ▪ C4100: unreferenced formal parameter; ▪ C4127: conditional expression is constant; ▪ C4244: 'conversion' conversion from 'type1' to 'type2', possible loss of data; ▪ C4267: conversion from Type1 to Type2: possible loss of data; ▪ C4189: local variable is initialized but not referenced; ▪ C4190: 'identifier1' has C-linkage specified, but returns UDT 'identifier2' which is incompatible with C ▪ C4244: conversion from 'X' to 'Y'. Possible loss of data; ▪ C4389: 'operator' : signed/unsigned mismatch; ▪ C4505: unreferenced local function has been removed; ▪ C4512: assignment operator couldn't be generated; ▪ C4701: potentially uninitialized local variable used; ▪ C4702: 'operator' : unreachable code; ▪ C4706: assignment within conditional expression. <p>Some other corrections have been introduced:</p> <ul style="list-style-type: none"> ▪ Old-style declarations of C functions have been modified in IGES and STEP parsers; ▪ Comparison is now used instead of assignment in BrepFill_TrimShellCorner; ▪ Wrong comparison of Boolean with index has been fixed in IntPatch_Align; ▪ Field IsCustomMatrix in InterfaceGraphic_Visual3d.hxx has become Boolean; ▪ Obsolete methods have been removed from TopOpeBRepBuild_ShapeSet.
--	---

23220	<p><i>Summary:</i> Porting training materials</p> <p>The training materials have been ported to the latest version of OCCT.</p> <p>The building procedures for Qt have been replaced by generation from pro files on Linux and Windows platforms.</p>
23955	<p><i>Summary:</i> Improve Cmake build scripts to make tests operable after install</p> <p>Cmake build scripts for OCCT now allow running tests on the built version without manual customization. There is a new option to install tests with binaries, in this case test scripts are copied to install directory and CASROOT can be set there.</p>
23957 24054	<p><i>Summary:</i> CmakeLists.txt file</p> <p>All static code for Cmake scripts is now contained in CmakeLists.txt located in OCCT root folder.</p> <p>Dynamic information generated by WOK (lists of files per toolkit, dependencies, etc) is still located in adm/cmake folder.</p>
23934	<p><i>Summary:</i> Compiler warnings in MSVC++ 2010</p> <p>The following modifications have been introduced to get rid of MSVC++ 2010 compiler warnings:</p> <ul style="list-style-type: none"> ▪ In ExprIntrp package: file ExprIntrp.tab.c and related WOK scripts have been removed; missing declarations and casts have been added in StepFile; warnings caused by Flex and Bison code have been suppressed by #pragma; ▪ In OSD package: dummy #includes have been added to files containing no code for Windows, to avoid warning on empty file; ▪ In Plib package: some code from Plib_ChangeDim.gxx has been included explicitly in cxx and cleaned; ▪ Additionally some casts have been added to avoid warnings.
24046 24048 24211	<p><i>Summary:</i> "Basic Runtime Checks" option</p> <p>"Basic Runtime Checks" option for VS 32-bit projects (except for VS 2012) has been changed to "RTC1". This allows capturing more errors at run time in Debug mode.</p> <p>The method IGESData_GlobalSection::Params has been refactored: the size of nombre array has been increased from 20 to 1024. The increased size allows avoiding the error of a float number conversion by Interface_FloatWriter::Convert.</p>
24162 24168 24169 24170 24171 24177	<p><i>Summary:</i> Eliminate Clang compiler warnings</p> <p>The OCCT code has been revised to eliminate the following warnings, shown by Clang compiler on Ubuntu 13.04 :</p> <ul style="list-style-type: none"> ▪ -Wlogical-op-parentheses (&& within) ▪ -Wunused-private-field ▪ -Wunused-variable ▪ -Wunused-value ▪ -Wunused ▪ -Wreorder



<p>24180 24236 24239 24274 24298 24304</p>	<p><i>Summary:</i> Eliminate GCC compiler warning</p> <p>The code has been modified to get rid of GCC compiler warnings about the following issues:</p> <ul style="list-style-type: none"> ▪ Multiline comments; ▪ Uninitialized variables; ▪ Wrong initialization order of class members; ▪ Functions unused in Linux; ▪ Switch statement on enumeration, where not all values are explicitly handled; ▪ Case label value exceeds maximum value for type.
<p>24227 24248</p>	<p><i>Summary:</i> Enable SSE2 instructions for OCCT building</p> <p>SS2 instructions have been enabled for OCCT building on 32-bit platforms to gain additional performance.</p>
<p>24234 24301 24436</p>	<p><i>Summary:</i> OCCT master is not compiled by VC++ 2005 (vc8 32/64 bit TKBO)</p> <p>The following modifications have been made to provide building by Visual Studio 2005 and 2013 (Express):</p> <ul style="list-style-type: none"> ▪ macro <code>_WIN32_WINNT</code> has been defined in <code>BOPCol_TBB.hxx</code> for building with TBB on VS 2005; ▪ compiler warnings specific for VS 2005 (vc8) have been fixed (e.g. explicit casts added in assignment expressions involving small integer types); ▪ definition of functions <code>atanh</code>, <code>acosh</code> and <code>asinh</code> is avoided for vc12 and later; ▪ math constants (<code>M_PI</code> and similar) are defined by including <code>math.h</code> rather than by a custom code. ▪ Support of Visual Studio 2013 has been provided in generation of projects by WOK
<p>24257</p>	<p><i>Summary:</i> Minor corrections for building / testing on VC++ 11</p> <p>The following corrections have been made to provide building and testing on MSVC++ 11:</p> <ul style="list-style-type: none"> ▪ Compiler warnings have been corrected in <code>structiges.c</code> (32-bit mode) and <code>BOPAlgo</code> (64-bit mode) ▪ <code>DRAW</code> command for testing exceptions (OCC6143) has been refactored to get some output even if the process fails; ▪ <code>DRAW</code> command <code>readstep</code> has been removed as it leads to hang-up in Debug mode due to waiting for user input; <code>testreadstep</code> command is used instead.
<p>24273</p>	<p><i>Summary:</i> Eliminate compiler warnings in <code>lex.ExprIntrp.tab.c</code> generated by flex 2.91</p> <p><code>ExprIntrp.lex</code> has been modified to avoid warnings generated by flex 2.91.</p>
<p>24275</p>	<p><i>Summary:</i> Cppcheck warnings on uninitialized class members</p> <p>Proper default initialization has been provided for uninitialized class members reported by Cppcheck.</p>



24284	<p><i>Summary:</i> Trivial warnings produced by ICC 14</p> <p>The OCCT code has been revised to eliminate the following warnings produced by ICC 14 compiler:</p> <ul style="list-style-type: none"> ▪ # 82: storage class is not first (e.g. “<code>int static f()</code>”); ▪ # 2621: attribute “<code>dllexport</code>” does not apply here (e.g. “<code>Standard_EXPORT typedef int qqint;</code>”); ▪ # 2415: variable ... of static storage duration was declared but never referenced; ▪ # 111: statement is unreachable (usually “<code>break</code>” after “<code>return</code>” in cycles).
24312	<p><i>Summary:</i> Compilation issues on OS X 10.9 Mavericks SDK</p> <p>The code has been modified to get rid of compilation issues on OS X 10.9 SDK (Xcode 5).</p>
24331 24426	<p><i>Summary:</i> Improvement of CMake scripts</p> <p>The following improvements have been introduced into CMake scripts:</p> <ul style="list-style-type: none"> ▪ Prefix <code>3RDPARTY</code> has been removed from the variables that trigger use of 3rd-party products. They have been moved to a separate group <code>USE</code>; ▪ Help strings associated with CMake variables have been revised; ▪ Tcl8.6 libraries now can be properly identified; ▪ Invariable CMake files have been moved from <code>wok</code> to <code>adm/templ ates</code> folder; ▪ Option <code>/fp: precise</code> has been explicitly set for Visual Studio projects, as a protection against use of the Intel compiler for which <code>/fp: fast</code> is a default one; ▪ <code>RELWITHDEBINFO</code> build configuration has been added; ▪ <code>BUILD_TYPE</code> variable has been renamed into <code>BUILD_CONFIGURATION</code>.
24377	<p><i>Summary:</i> OCC 6.7.0 beta contaminates log with unnecessary debug symbols</p> <p><code>DEBUG</code> outputs have been replaced with corresponding comments in <code>Extrema_FuncExtCC</code> and <code>Extrema_FuncExtPC</code> classes.</p>
24399	<p><i>Summary:</i> ICC warnings 3280 “declaration hides...”</p> <p>Several dozen local variables that have the same name as a class field or another variable in the same scope have been renamed to avoid ICC compiler warnings.</p>

Samples

23459	<p><i>Summary:</i> Update of QT samples</p> <p>The following modifications have been introduced in the frame of samples update:</p> <ul style="list-style-type: none"> ▪ QT samples have been enabled work with Qt4 built without Qt3 support. ▪ New sample Voxel Demo. pro has been added to illustrate the usage of voxels. ▪ The building procedures for qt have been replaced by generation from pro files on Linux and Windows platform
23874	<p><i>Summary:</i> Convert OCCT MFC samples to CMake build system.</p> <p>CMake system has been implemented for building MFC samples instead of multiple VS projects.</p>
23931	<p><i>Summary:</i> Incorrect image export code in MFC sample</p> <ul style="list-style-type: none"> ▪ The methods OCC_BaseDoc: : SupportedImageFormats() and OCC_BaseDoc: : ExportView() have been implemented to unify the export procedure in all MFC samples. ▪ OnFileExportImage() events have been corrected in BestFit, CollisionDetection, MeshFW, ShapeHealing and XDE samples.
24147	<p><i>Summary:</i> Update of CSharp samples</p> <p>The code of CSharp samples has been refactored. The following modifications have been introduced in the frame of samples update:</p> <ul style="list-style-type: none"> ▪ Project files corrected for samples to run on 64-bits systems; ▪ WPF front end has been added to demonstrate usage of OCCT in WPF applications; ▪ Option "Export image to .xwd" has been removed.
24355	<p><i>Summary:</i> Compiler Warning level 4 for MFC samples</p> <p>MFC samples have been upgraded to warning level 4 (with MSVS compiler).</p>
24372	<p><i>Summary:</i> HLR Sample: erased objects are shown in HLR</p> <p>The following improvements have been introduced in HLR sample:</p> <ul style="list-style-type: none"> ▪ HLR view is now updated after objects are erased from the 3D view. ▪ HLR option has been added to menu "File". ▪ The information about sample has been added to README file and is now displayed when "Help" button is pressed in the toolbox or in the menu.
24399	<p><i>Summary:</i> Vectors not displayed correctly in MFC samples [6.7.0 Beta]</p> <p>ISession_Direction classes that provide visualization of directions have been rewritten without using DsgPrs_LengthPresentation class, as its default value of arrow length differs from the one in drawer's arrow aspect.</p> <p>Method ISession_Direction: : SetLineAspect() has been added to support old results in geometry samples.</p>

Release

23977	<p><i>Summary:</i> Make packaging of OCCT more uniform</p> <p>3rdparty subfolder has been removed. OCCT and 3rd party products are now located at the same directory level in the installed OCCT package.</p>
24197	<p><i>Summary:</i> Viewer3d.exe is saved to a wrong directory</p> <p>The output directory for Viewer3d.exe has been corrected.</p>

Documentation

23962 24269 24336	<p><i>Summary:</i> Moving OCCT documentation to sources</p> <p>OCCT user documentation (Overview and User Guides) has been converted from MS Word files to text-based format and added to OCCT sources, in the new subfolder <code>dox</code>. HTML and PDF pages can be generated from the sources using <code>tcl</code> and <code>bat</code> scripts with <code>Doxygen</code> and <code>MiKTeX</code> as prerequisites.</p>
24330	<p><i>Summary:</i> Generate Overview documentation in CHM format</p> <p>It has become possible to generate user documentation in CHM format using <code>Doxygen</code>, which can be convenient for Windows users.</p>
24341	<p><i>Summary:</i> Documentation about building OpenCL ICD Loader package</p> <p>The building of OpenCL ICD Loader has been described in user's documentation. Descriptions of <code>CMake</code> and <code>Automake</code> building procedures have been updated.</p>
24351	<p><i>Summary:</i> Create Coding Rules document</p> <p>An instruction about Coding Rules applicable to the development of OCCT has been added in the documentation.</p>
24361	<p><i>Summary:</i> Enable brief documentation of class members in detailed description</p> <p>Brief description of class members has been enabled in class member list in OCCT reference manual.</p>

ProductsExpress Mesh

22791	<p><i>Summary:</i> Shapes imported from IGES are meshed with holes</p> <p><code>QMShape_Tessellator</code> has been modified to not take into consideration edges with 2d and 3d curves of zero length, which avoids failures in the algorithm.</p>
23980	<p><i>Summary:</i> <code>Standard_NumericError</code> during face meshing</p> <p>Method <code>QMShape_Tessellator::Perform()</code> has been protected against corrupted degeneracy flag.</p>
24006	<p><i>Summary:</i> Command <code>QMdiscr</code> causes access violation if options <code>-export -parallel</code> are used</p> <p>The syntax of <code>-parallel</code> option for command <code>QMdiscr</code> has been simplified. Now sequential mode is used by default, and parallel mode is activated by <code>-parallel</code> option with no arguments.</p>
24014	<p><i>Summary:</i> Express Mesh fails division procedure for polygon with "glued" neighboring links</p> <p>Glued links in <code>Polygon2dTool</code> have been replaced with a single link connecting the start node of the first link and the last node of the second one. Additionally:</p> <ul style="list-style-type: none"> ▪ Both distance and intersection are now checked instead a single check by angle; ▪ The final point is correctly identified after removing a loop.
24034	<p><i>Summary:</i> Incorrect mesh produced</p> <p>The following improvements have been introduced in classes <code>QMTools_Polygon2dTool</code> and <code>QMBgr_FacetBuilder</code>:</p> <ul style="list-style-type: none"> ▪ Ratio factor is not scaled if a node that is not contained in the map of preferred nodes is chosen; ▪ Synchronization of polygon division in 2d with 3d space is additionally checked to avoid overlapping areas; ▪ Pseudo-parametric (projected) polygons are used instead of real parametric ones for calculation of the best link in 3D space.
24152 24173	<p><i>Summary:</i> Express mesh produces a bad mesh on sphere</p> <p>The following improvements have been introduced to improve meshing of spheres:</p> <ul style="list-style-type: none"> ▪ In <code>QMBgr_FacetBuilder</code> class the coordinate provider is switched to a default coordinate in case of failure on degenerative edges; ▪ Method <code>QMTools_Polygon2dTool::RemoveSmallLoops</code> has been improved to leave the source polygon unchanged and return the updated polygon via out parameter. ▪ Contour finding algorithm <code>QMBgr_FacetBuilder::FindContour</code> has been extended with walking behavior. It allows rolling back to the previously chosen connection and selecting another way in case of a hanging end. ▪ In the same class, connections between the node binding map and the resulting contour have been removed to avoid producing overlapped facets.

24237	<p><i>Summary:</i> Express Mesh produces non-connected mesh</p> <p>The method <code>QMBgr_FacetBuilder::IsConvex</code> has been modified to avoid checking the facet convexity if the target element is a triangle.</p>
-------	--

Advanced Samples and Tools

23583	<p><i>Summary:</i> Add the possibility to switch between sequential / parallel meshing in OMF sample</p> <p>The necessary controls have been added to allow switching between sequential / parallel meshing in OMF sample.</p>
24306	<p><i>Summary:</i> Problems in OCCT training exercises</p> <p>OCAF and Shape Healing training exercises have been revised and corrected.</p>

DXF Import-Export

22396	<p><i>Summary:</i> Annotations cannot be read from DXF</p> <p>It has become possible to read text entities from DXF file into XCAF documents. They can also be visualized at the level of XDE.</p>
23986	<p><i>Summary:</i> The problem with write+read into DXF-file</p> <p><code>DxfFile</code> class has been modified to allow writing shapes into DXF format files without loading an additional plugin or reading from file before writing.</p>
24116	<p><i>Summary:</i> Extended data are not read from a DXF file</p> <p>New class <code>XAppData</code> has been added instead of <code>myXData</code> in <code>HandledObject</code> to correct reading of DXF files, which contain more than one group code 1001 (Application name) attached to one entity.</p> <p><code>RWHandledObject</code> and <code>RWXdata</code> have been changed to allow reading <code>XData</code> which does not contain first open brackets (group code 1002), as according to DXF Reference, this group code is not necessary.</p>
24373	<p><i>Summary:</i> Invalid shape translating a DXF file</p> <ul style="list-style-type: none"> ▪ Reading of subtypes of <code>Aci sGeom_IntCurve</code> has been corrected; ▪ Initialization of a map of types has been protected by mutex.

ACIS-SAT Import-Export

24082	<p><i>Summary:</i> Exception translating a SAT file</p> <p>The algorithm reading fields has been corrected in method <code>Aci sGeom_IntCur::SetData</code>.</p>
-------	--

PARASOLID-XT Import

23973	<p><i>Summary:</i> Invalid shape after importing an XT file</p> <p>The transformation applied to a 2d curve on spun surface in <code>XtToGeom_MakeCurve2d: : Transform2dCurve</code> has been corrected.</p>
24020	<p><i>Summary:</i> Invalid shape after importing an XT file</p> <p><code>XtToGeom_MakeBlendedEdge</code> class has been modified to enable constructing <code>BlendedEdge</code> as a primitive (like a cylinder or a conical surface), if its <code>range1</code> and <code>range2</code> are null.</p> <p><code>XtToTopoDS_TranslateFin</code> has been modified to correct translation in some files, which contain edges with condition $(U1 - U2) < PConfusion$, where <code>U1</code> and <code>U2</code> are parameters of the curve lying on this edge.</p>
24395	<p><i>Summary:</i> OCC License protection is not thread safe</p> <p><code>OCCLicense</code> and <code>OCCLicenseTools</code> classes that perform license checking have been made thread safe.</p>

Products Building

24338 24409	<p><i>Summary:</i> Eliminate compiler warnings in OCC Products</p> <p>The code of Products has been revised to get rid of compiler warnings.</p>
----------------	--

Porting to version 6.7.0

Porting of user applications from the previous 6.6.0 OCCT version to version 6.7.0 requires the following issues to be taken into account:

Object-level clipping and capping algorithm.

- It might be necessary to revise and port code related to management of view-level clipping to use `Graphic3d_ClipPlane` instead of `V3d_Plane` instances. Please note that `V3d_Plane` class has been preserved - as previously, it can be used as plane representation. Another approach to represent `Graphic3d_ClipPlane` in view is to use custom presentable object.
- The list of arguments of `Select3D_SensitiveEntity::Matches()` method for picking detection has changed. Since now, for correct selection clipping, the implementations should perform a depth clipping check and return (as output argument) minimum depth value found at the detected part of sensitive. Please refer to CDL / Doxygen documentation to find descriptive hints and snippets.
- `Select3D_SensitiveEntity::ComputeDepth()` abstract method has been removed. Custom implementations should provide depth checks by method `Matches()` instead - all data required for it is available within a scope of single method.
- It might be necessary to revise code of custom sensitive entities and port `Matches()` and `ComputeDepth()` methods to ensure proper selection clipping. Please note that obsolete signature of `Matches` is not used anymore by selector. If your class inheriting `Select3D_SensitiveEntity` redefines the method with old signature the code should not compile as the return type has been changed. This is done to prevent override of removed methods.

Redesign of markers presentation

- Due to the redesign of `Graphic3d_AspectMarker3d` class the code of custom markers initialization should be updated. Notice that you can reuse old markers definition code as `TColStd_HArray1ofByte`; however, `Image_Pixmap` is now the preferred way (and supports full-color images on modern hardware).
- Logics and arguments of methods `AIS_InteractiveContext::Erase()` and `AIS_InteractiveContext::EraseAll()` have been changed. Now these methods do not remove resources from `Graphic3d_Structure`; they simply change the visibility flag in it. Therefore, the code that deletes and recomputes resources should be revised.
- `Graphic3d_Group::MarkerSet()` has been removed. `Graphic3d_Group::AddPrimitiveArray()` should be used instead to specify marker(s) array.

Default views are not created automatically

As the obsolete `Init()`, `DefaultOrthographicView()` and `DefaultPerspectiveView()` methods have been removed from `V3d_View` class, the two default views are no longer created automatically. It is obligatory to create `V3d_View` instances explicitly, either directly by operator `new` or by calling `V3d_View::CreateView()`.

The call `V3d_View::SetDefaultLights()` should also be done explicitly at the application level, if the application prefers to use the default light source configuration. Otherwise, the application itself should set up the light sources to obtain a correct 3D scene.



Improved dimensions implementation

- It might be necessary to revise and port code related to management of `AIS_LengthDimension`, `AIS_AngleDimension` and `AIS_DiameterDimension` presentations. There is no more need to compute value of dimension and pass it as string to constructor argument. The value is computed internally. The custom value can be set with `SetCustomValue(...)` method.
- The definition of units and general aspect properties is now provided by `Prs3d_DimensionUnits` and `Prs3d_DimensionAspect` classes.
- It might be also necessary to revise code of your application related to usage of `AIS_DimensionDisplayMode` enumeration. If it used for specifying selection mode, then it should be replaced by a more appropriate enumeration `AIS_DimensionSelectionMode`.

NCollection Set replaced by List collection

It might be necessary to revise your application code, which uses non-ordered `Graphic3d_SetOfHClipPlane` collection type and replace their occurrences by ordered `Graphic3d_SequenceOfHClipPlane` collection type.



Supported Platforms and Pre-requisites

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

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

The most up-to-date information on Supported Platforms and Pre-requisites is available at <http://www.opencascade.org/getocc/require/>.

Linux Operating System	Mandriva 2010, CentOS 5.5, CentOS 6.3, Fedora 17, Fedora 18, Ubuntu-1304, Debian 6.0 *
Windows Operating System	MS Windows 8 / 7 SP1 / Vista SP2 / XP SP3
Mac OS X Operating System	Mac OS X 10.9 Mavericks / 10.8 Mountain Lion / 10.7 Lion / 10.6.8 Snow Leopard
Minimum memory	512 MB, 1 GB recommended
Free disk space (complete installation)	650 MB of disk space, or 1,4 GB if installed with reference documentation
Graphic library	OpenGL 1.1+ (OpenGL 2.1+ is recommended)
C++ <i>For Linux:</i> <i>For Windows:</i> <i>For Mac OS X:</i>	GNU gcc 4.0. - 4.7.3. Microsoft Visual Studio 2005 SP1 with all security updates Microsoft Visual Studio 2008 SP1** Microsoft Visual Studio 2010 SP1 Microsoft Visual Studio 2012 Update 3 Microsoft Visual Studio 2013 Intel C++ Composer XE 2013 SP1 XCode 3.2 or newer (4.x is recommended)
TCL (for testing tools) <i>For Linux:</i> <i>For Windows:</i> <i>For OS X:</i>	Tcltk 8.5 or 8.6 http://www.tcl.tk/software/tcltk/8.6.html ActiveTcl 8.5 or 8.6 http://www.activestate.com/activetcl/downloads Built-in Tcl/Tk 8.5
Qt (for demonstration tools)	Qt 4.6.2 http://qt.nokia.com/downloads
Freetype (OCCT Text rendering)	freetype-2.4.11 http://sourceforge.net/projects/freetype/files/
FreeImage ** (Support of common graphic formats)	FreeImage 3.15.4 http://sourceforge.net/projects/freeimage/files/Source%20Distribution/
gl2ps ** (Export of OCCT viewer contents to vector graphic file)	gl2ps-1.3.8 http://geuz.org/gl2ps/
TBB (optional tool for multithreaded algorithms)	TBB 3.x or 4.x http://www.threadingbuildingblocks.org/
OpenCL (optional for ray tracing visualization core)	OpenCL 1.2.8 (with GPU devices for run-time Ray Tracing rendering)
Doxygen (optional for building documentation)	Doxygen 1.8.5 http://www.stack.nl/~dimitri/doxygen/download.html

- * Debian 60 64 bit is a permanently tested platform.
- ** The official release of OCCT for Windows contains libraries built with VC++ 2008.