Open CASCADE Technology and Products ver. 6.5.1 Maintenance Release

Release Notes

Overview

Open CASCADE Technology and Products version 6.5.1 is a maintenance release, which includes **57** new features, improvements and bug fixes, over minor release 6.5.

Version **6.5.1** is binary incompatible with the previous versions of Open CASCADE Technology and Products, so applications linked against a previous version must be recompiled to run with this Version 6.5.1.

Highlights

- **New tools for memory usage analysis and optimization**
- **Improved performance and correctness of `BRepMesh` algorithm**
- **Optimization of point-on-surface projection in `Extrema` package**
- **Source code of DRAW commands included in the Products delivery**
- **Multiple improvements to increase robustness**

Table of Contents

Modifications

Foundation Classes

22143	*Summary:* Optimization of memory usage and developing tools for catching memory leaks in OCCT-based applications The following improvements have been introduced to optimize memory usage: ▪ The callback defined in the class `NCollection_BaseAllocator` now allows reporting the numbers of active memory blocks, which simplifies the detection of memory leaks. This callback can be activated by the global method `Standard_Size& StandardCallBack_CatchSize()`, which defines the size of blocks to be included in the report. The global method `Standard_Size& StandardCallBack_CatchID()` allows setting a breakpoint inside the callback at the reserved statement when the block with a certain allocation number is allocated or freed. Finally, it is possible to reset the callback into the initial state using the global method `void StandardCallBack_Reset()`. The report is now output to the file `memstat.d` instead of the standard output. ▪ New mechanism to detect alive instances of the class `NCollection_IncAllocator` has been added. It works in debug mode only (DEB preprocessor variable is defined). To activate it, use the global method `void IncAllocator_SetDebugFlag(const Standard_Boolean theDebug)`. With this flag is on, each call to constructor or destructor of the class `NCollection_IncAllocator` registers or unregisters the instance. At some point, it is possible to call the global method `void IncAllocator_PrintAlive()`, which writes the file "`inc_alive.d`" with information about alive instances of the allocator and the size occupied by them. Additionally, the following changes have been made in the frame of the optimization: ▪ An error breaking the consistency of `NCollection_List` object if the method `Prepend()` is called on an empty list has been fixed. ▪ It has become possible to use the allocator from the Constructor of `NCollection_Vector` class not only to allocate the table of block headers, but also to allocate arrays of items. So, if the allocator is used for a vector, no memory is allocated for vector purposes outside of the given allocator. ▪ A dedicated constructor has been added to create a handled tree with allocator in `NCollection_UBTree`.
22256	*Summary:* Add mechanism based on `malloc/free` callback for debugging memory problems It has become possible for the user to set a custom callback to process memory allocation events. There are two ready to use callbacks that allow to generate the report in the file in tabular form providing statistics for allocation/deallocation of each block size. This mechanism has been implemented via the new class `OSD_MAllocHook` and the corresponding Draw command "`mallochook`". Attention: this version works only on Windows platform in Debug build mode. It relies on the debug CRT function `_CrtSetAllocHook` (see MSDN for help).

22273	*Summary:* `Message_ProgressSentry` with NULL handle cause exception `Message_ProgressSentry.lxx` has been fixed to avoid wrong check performed to the handle `(not ::IsNull() method)`.
22278	*Summary:* No sorting algorithm implemented for NCollection templates The classic `QuickSort` algorithm has been implemented for commonly used templates, such as `NCollection_Sequence` and `NCollection_Vector`.
22301	*Summary:* `TColStd_PackedMapOfInteger` issue with extent value The problem with incorrect value of `Extent()` returned after some operations has been corrected in class `TColStd_PackedMapOfInteger`.
22355	*Summary:* Avoid annoying warnings in `NCollection_SparseArray.hxx` on 64-bit Linux with Intel compiler `NCollection_SparseArrayBase.hxx` has been modified to avoid compilation warnings.
22360	*Summary:* Writing out of allocated memory in the method `OSD_FontMgr::InitFontDataBase` Method `OSD_FontMgr::InitFontDataBase` has been corrected to allocate enough memory for string storage in the variable `windir_var`.

Modeling Data

22400	*Summary:* `BndLib_Add3dCurve::Add` – protect against void bounding box The method `BndLib_Add3dCurve::Add` has been protected against exception in case if a void bounding box is generated for a `bspline` curve.

Modeling Algorithms

22138	*Summary:* Remove `*.gxx` files from Mesh algorithm `MeshDS` and `MeshAlgo` packages have been fully removed and all their structures have been moved to `BrepMesh` package. Generic classes (`*.gxx` files) have become concrete implementations. Names of structures have been changed in accordance with the existing `BrepMesh` aliases.
22139	*Summary:* Statuses of `BrepMesh` Method `GetStatus` that returns the current meshing status has been added into `BrepMesh_IncrementalMesh.`

22145	*Summary:* Incorrect cutting `BRepMesh_FastDiscretFace` The mechanism which restores Delaunay structures in `BRepMesh` package has been improved. The benefit of this change consists in code duplication removal and decrease of the number of math calculations. The following changes have been implemented in the framework of this improvement: ▪ Number of parameters in the Constructor of `BRepMesh_FastDiscretFace` class has been changed. ▪ `BRepMesh_FastDiscretFace::RestoreStructureFromTriangulation` method has been introduced to restore Delaulay structure from the existing face triangulation without unnecessary data recalculation (therefore method `Update` has been removed). ▪ The following unnecessary members have been removed: `myInternalVerticesMode` from `BRepMesh_FastDiscret`; `edges, myshapetrigu, myinshape` from `BRepMesh_FastDiscretFace`; ▪ Method `FindUV` has become static in `BRepMesh_FastDiscretFace` and `BRepMesh_FastDiscret` calls it from there. New parameter, which is a reference to the map of `DataMapOfIntegerListOfXY` for storing results of calculations, has been added in this method. Declaration and implementation of `FindUV` has been removed from `BRepMesh_FastDiscret`. Additionally, static modifiers have been removed from method `BRep_Tool::CurveOnSurface` to provide correct access to the resources from various threads when parallelization is used.
22163	*Summary:* Creation of non-conformal quadratic pyramids `ChFi3d_Builder_0.cxx` and `TopOpeBRepBuild_BuildFaces.cxx` have been modified to avoid creating mesh pyramids too close to each other.
22247 22342	*Summary:* `BRepMesh` poor performance Performance of `BRepMesh_FastDiscretFace` has been optimized on NURBS surfaces. Robustness and performance of the face interior computation algorithm have been improved.
22286	*Summary:* Intersection between two faces gives different results. Method `IntTools_FaceFace::Perform` has been modified to prevent the dependence of the result of intersection between two faces `(F1, F2)` upon the order of faces: `[F1,F2]` or `[F2,F1]`.
22303	*Summary:* The algorithm `BRepExtrema_DistShapeShape` crashes with exception if one of the shapes does non contain vertices Loops "`do {...} while(...)`" have been replaced by loops "`while(...) {...}`" in `BRepExtrema_DistShapeShape.cxx` to avoid exception.

22311	*Summary:* A regression was found: face obtained from 2D offset of a wire is invalid The following changes have been introduced to address this issue: ▪ New generated class `IndexedDataMapOfOrientedShapeListOfShape` has been added to `BRepFill.cdl.` ▪ The field "myMap" of class `BRepFill_OffsetWire` changes its type to `BRepFill_IndexedDataMapOfOrientedShapeListOfShape`. ▪ Usage of `IndexedDataMap` instead of `DataMap` in some cases makes the algorithm more stable. ▪ The edge orientation is now preserved in BRepFill_OffsetWire.cxx before substitution of its vertices to prevent loss of right orientation. ▪ The algorithm in `MAT_Mat.gxx` has been made more simple and clear.
22318	*Summary:* Regression (since OCCT 6.3.1): bad shading of small cylinder `BRepMesh_FastDiscret` algorithm has been changed so that the triangulator could take into account the type of edge. It the edge is circle-like, the number of vertices can be more than two. This improvement has been implemented to address problems with meshing of a compound consisting of small and big cylinders.
22322	*Summary:* Improvement of `Extrema` performance The new rapid algorithm `Extrema_ExtAlgo_Tree` which finds minimum and maximum distances between point and surface using B-tree (note that b-tree finds only one point!) has been implemented in package `Extrema`. For backward compatibility, the old algorithm `Extrema_ExtAlgo_Grad` which uses gradient descent to find local extreme points on surface is still available and is used by default. The algorithm can be set using method `::setAlgo(Extrema_ExtAlgo)`. On the other hand, it has become possible to search in different modes: for minimum, maximum or both. The mode can be set using method `::setFlag(Extrema_ExtFlag)`. Possible modes are `Extrema_ExtFlag_MIN`, `Extrema_ExtFlag_MAX` or `Extrema_ExtFlag_MINMAX`. The algorithm and mode can be set in `BRepExtrema_ExtPF` or in low-level `Extrema_ExtPS` and `Extrema_GenExtPS` classes. `Extrema` classes are used in `BRepExtrema` as class fields instead of local variables to cache results, so `Extrema` should be reinitialized after the algorithm is changed, otherwise the search would fail.
22324	*Summary:* Mistakes with parenthesis position in abs calls Position of parentheses has been corrected in `abs` calls of the classes `Adaptor3d_CurveOnSurface`, `Aspect_RectangularGrid` and `GeomFill_BoundWithSurf`.
22329	*Summary:* Skipped extrema for a point and a circle at about 0 parameter `Extrema_ExpPElC.cxx` has been modified to find a minimum distance between a point and a circle if the point is located near zero parameter of the circle.
22361	*Summary:* Incorrect result of `BRepOffsetAPI_MakePipeShell` algorithm: it tries to build conical surface between two non-coaxial circles New field `"myTrsfs"` has been added to classes `BRepFill_NSections` and `GeomFill_NSections` to keep information about the original disposition of sections.

22401	*Summary:* Disable debug printouts in `GeomPlate_BuildPlateSurface` algorithm Useless debug printouts have been disabled in the algorithm `GeomPlate_BuildPlateSurface`.
22428	*Summary:* The shape is valid on Linux but non-valid on Windows. `BRepCheck_Edge::Standard_BooleanValidate` has been modified to prevent the loss of accuracy of floating-point arithmetic for ellipse and cone.
22503	*Summary:* Face triangulation causes shading display of whole shape to fail Class `Poly_Connect` has been enabled to work correctly on triangulations with free nodes. DRAW command `checktopo` now can identify triangulations containing free nodes as erroneous

Visualization

22105	*Summary:* Suspicious call to `XOpenDisplay()` in `OSD_FontMgr` class Application using 3D viewer through a remote connection to server might crash because of calls to `XOpenDisplay()` in `OSD_FontMgr` class. The following changes have been made to address this issue: ▪ `OSD_FontManager::InitFontDataBase` method now tries to connect to client's display with `XOpenDisplay("localhost:0.0")` call at first, and then, if it is failed, with `XOpenDisplay(":0.0")` call. ▪ Uncontrolled memory allocation, which occurred if the font database of `OSD_FontManager` was empty, has been corrected in `OpenGl_TextRender::RenderText` method.
22108	*Summary:* Cutting plane unpredictable behavior in `V3d_View` The behavior of clipping planes has been modified in `OpenGl` package. Now clipping planes are updated when the viewer's content is redrawn.
22144	*Summary:* NIS performance and memory usage update NIS package has been modified and optimized in the following way: ▪ The allocators to keep in memory a huge number of NIS objects have been introduced. ▪ Memory optimization has been implemented for drawing of Open GL draw lists with a small (one byte) or average (two bytes) number of objects. ▪ Draw list update functions have been optimized.
22150	*Summary:* Problem with `GetWindowLong` function on 64-bit platform The instability problem caused by the function `SetWindowLong` and `GetWindowLong` in case of high (3-8 Gb) memory amount taken by the application has been solved by substituting calls to `SetWindowLong` by `SetWindowLongPtr` and calls to `GetWindowLong` by `GetWindowLongPtr`.

22199	*Summary:* OpenGL memory leaks in TKOpenGl The OpenGl resource handling mechanism has been modified. Now the video memory resources are cleaned up when the OpenGl driver can handle this operation (before redrawing). The following changes have been introduced into packages: ▪ Shading aspect initialization for the textured shape has been corrected in the AIS_TextureShape::Compute method. Now the reference counter of a texture is handled correctly. ▪ The Constructor of Graphic3d_ArrayOfPrimitives class has been corrected to initialize the value of primitive context. So the primitive array remembers the OpenGl context when the corresponding VBOs are created for the array. ▪ The OpenGl context identifier field has been added to the CALL_DEF_PARRAY structure in package InterfaceGraphic. ▪ OpenGl_ResourceCleaner class has been added to handle the resource cleaning on redraw; ▪ OpenGl_Resource, which is now the base class for any video resource handled by OpenGl_ResourceCleaner, has been added; ▪ OpenGl_ResourceVBO class has been added to handle VBO video memory resources; ▪ OpenGl_ResourceTexture class has been added to handle the texture video memory resources; ▪ The list of input parameters of "vdrawsphere" ViewerTest command has been modified. Now it is necessary to use: vdrawsphere ShapeName Fineness [X Y Z] [Radius] [EnableVBO] [NumberOfViewerUpdate] [ShowEdges]. It has become possible to use this command for memory leak testing; ▪ "vtexture" command algorithm has been corrected. Now it is possible to use this command for memory leak testing.
22272	*Summary:* Small improvements in selection and presentation Method Graphic3d_Structure::Groups() has been changed. Previously it returned the filled Handle(Graphic3d_HSetOfGroup) collection each time, which was not optimal for frequent access. This method has been redefined to return the const Graphic3d_SequenceOfGroup& as stored in the class.

22354	*Summary:* Bug in Overlay Text rendering
	
	It has become possible to specify for text rendering if the text is 2D or 3D (with depth coordinate). 2D text cannot be hidden under a primitive on an overlay plane.
	The following changes have been introduced in OpenGl package in the frame of this improvement:
	▪ OpenGl_TextRender::RenderText method has been modified to specify for the OpenGl_FontMgr if the text is 2D or 3D. Now this method keeps texture state, so it has no changes made by FTFont after text rendering.
	▪ render_text methods from OpenGl_FontMgr have a new parameter "is2d", for example:
	`void render_text(const char* text, const Standard_Boolean is2d = 0);`
	`void render_text( const Standard_Integer id,  const char* text,`
	`                  const Standard_Boolean is2d = 0 );`
	These methods now support depth testing for 2d text correctly.
	▪ OpenGl::call_togl_redraw_layer2d method has been corrected to keep texture state bits unchanged.

22357	*Summary:* Hidden face selection The procedure of picking by point if custom clipping planes are defined has been improved. Previously the objects hidden behind the clipping were selectable, which made the selection of visible objects impossible or too complicated. Now depth clipping has been implemented for picking by point functionality, so hidden objects or their parts should not be selectable. The following changes have been introduced in the framework of this improvement: ▪ `Select3D_Projector` class now inherits `Standard_Transient`. This means that in most cases and existing methods it should be used as `Handle` with argument `Select3D_Projector` replaced by `Handle(Select3D_Projector)`. This has been done to avoid unsafe memory usage when deleted projector is possibly used in sensitivity entities and other places. ▪ New methods `::DepthMin()`, `::DepthMax()` and `::DepthMinMax()` have been added in `Select3D_Projector` class to control the allowed depth interval. These methods are now used in `Select3D_SensitiveEntity` and inheritors within `::Matches()` method (for picking in a point). ▪ `Select3D_SensitiveEntity.cdl` has been updated to operate with `Handle(Select3D_Projector)`. Some other classes have been automatically updated without CDL changing. ▪ Constructor of class `V3d_Plane` has been changed to not require the `V3d_Viewer` argument. Attention: the plane doesn't append itself to the list of planes defined in `V3d_Viewer`. ▪ Methods `V3d_Viewer::AddPlane()` and `V3d_Viewer::DelPlane()` have become public. Now the application controls the list of defined planes if needed. Additionally, the problem with some objects remaining highlighted when they should not be highlighted according to the current cursor position has been fixed in `TKOpenGl` though removal of obsolete low-level optimizations.
22362	*Summary:* `AIS_LocalContext` should be public class Some improvements have been introduced in AIS package to allow querying the current `LocalContext` instance, which helps to improve the performance when many highlighted objects should be processed in a loop: ▪ Class `AIS_LocalContext` has become public, it can be obtained by new method `AIS_InteractiveContext::LocalContext()`. ▪ Methods `AIS_LocalContext::FindSelectedOwnerFromIO()` and `AIS_LocalContext::FindSelectedOwnerFromShape()` have become public.
22377	*Summary:* Patch for visualization component Pre-overlay call backs have been implemented via the following modifications in OpenGl package: ▪ Pre-overlay callbacks have been added to `redraw` methods. The corresponding flag is `OCC_REDRAW_WINDOW | OCC_PRE_OVERLAY` for redrawing the whole window and `OCC_REDRAW_WINDOWAREA | OCC_PRE_OVERLAY` in case of redrawing a window area. ▪ The algorithm handling `glx` shared contexts has been corrected. ▪ Incorrect `"if"` statement has been modified in `OpenGl_attri::TsmPopAttri()` method. ▪ Shared contexts are now taken into account during the deletion of a viewer background texture with help of `OpenGl` resource cleaner.

22385	*Summary:* `AISInitViewer` Draw command creates too small viewer on Windows. `ViewerTest_Tool.cxx` has been modified to create a greater-sized viewer.
22391	*Summary:* Polylines arrays not drawn for mesh with `VBO` flag enabled The problem with polylines array drawing has been corrected. The drawing algorithm of `OpenGl_PrimitiveArray` has been modified to support polylines array with `VBO` flag enabled.
22443	*Summary:* Missing "`return`" statement in `WNT_Window::IsMapped()` "`return`" statement has been added to a control path in `WNT_Window::IsMapped()`.
22529	*Summary:* `FitAll` works incorrectly for small flat shapes Regression in `V3d_View::FitAll` has been fixed. Previously the scene was not zoomed enough to fit small flat shapes. The smaller the shape, the worse was the result.

Application Framework

18056	*Summary:* Exception during copying `Array` attribute with `array(0,0)` Several improvements have been introduced into `TDataStd` package to avoid `Access` violation exception during copying `Array` attribute with internal array `(0,0)`. The exception was caused by `NULL` handle instead of internal array in the field of target attribute.
22382	*Summary:* List of Undo/Redo should be protected A small modification has been introduced to permit the document transaction model to automatically track the changes. This is useful for proper updating of presentations and depending data objects. ▪ Virtual method `TDocStd_Document::CommitTransaction()` can be redefined so that the last delta (`myUndos.Last()`) could be iterated to take into account each attribute delta after successful commit. ▪ The same thing should be done in operations `Undo` and `Redo` before running a transaction: iterate `myRedos.First()` before calling `Redo()` and `myUndos.Last()` before calling `Undo()`.
22315	*Summary:* Correction of pure misprint in `TDatsXtd` & `ViewerTest` Misprints have been fixed in files `TDataXtd_Constraint.cxx` and `ViewerTest.cxx`.

Data Exchange

22238	*Summary:* Bug during step file importation `STEPControl_ActorRead` has been improved to import two specific STEP files correctly.
22294	*Summary:* Problem with reading of attached file with OCCT642 Transformation for spline curves has been added in method `IGESToBRep_TopoCurve::TransferTopoBasicCurve`.
22300	*Summary:* Debugging information is dumped in optimized version by STL reading API Debug dump to standard output (console) in STL reader has been eliminated.
22305	*Summary:* XDE Xml reader does not check the version of xml file Method `XmlLDrivers_DocumentRetrievalDriver::ReadFromDomDocument` has been modified so that the Xml document reader could check the document version for compatibility with the reader version. If the file version is greater than reader version, the reader status sets to `PCDM_RS_NoVersion` and no operation is performed.
22403	*Summary:* Avoid long recursion in the method `Transfer_Binder::CutResult` A recursion has been replaced with a cycle in the method `Transfer_Binder::CutResult`.
22504	*Summary:* Regression in 6.5.0: Some faces are missing after reading IGES `IGESToBRep_IGESBoundary.cxx` has been modified to avoid regression resulting in faces missing after translation.

Draw

22250	*Summary:* Windows lost focus with DRAWEXE launched in 'virtual windows' mode Since OCCT 6.4.2 DRAWEXE does not show TK/TCL and visualization windows in batch mode (launched with `-f Script.tcl` or with `–v` flag), but it is still possible to dump them with `vdump, v2ddump` and `xwd` commands. A nuisance has been noticed: while windows are not displayed, their focus is lost by the current system window (not OCCT). This is inconvenient when the workstation is used for regular work and not dedicated to perform only tests. This problem has been solved for Windows systems by a fix in `Draw_Window.cxx`.
22291	*Summary:* Improve `meshcolors` command to support texture interpolation DRAW command `meshcolors` from `XSDRAWSTLVRML` package has been improved to support texture-based color interpolation through an extra command-line option. The improved command can illustrate the use of `MeshVS_NodalColorPrsBuilder` class in the mode when texture mapping is used to display color interpolation across mesh elements. The following changes have been introduced in packages: ▪ `MeshVS_NodalColorPrsBuilder::Build` method has been modified to support lighting for texture interpolated mesh. Previously normals were not computed in case of using texture map for interpolation. Now, normals are computed in the same way as for colored mesh without textures. ▪ The initial texture parameters of Constructor of class `MeshVS_ImageTexture2D` have been changed to produce full colored texture on the mesh. Now this method uses `Modulation OpenGl` algorithm to compute the resulting mesh color. ▪ DRAW command `meshcolors` has been modified in `XSDRAWSTLVRML`. The new `"nodaltex"` mode argument builds texture interpolated mesh using the `MeshVS_NodalColorPrsBuilder`.
22418	*Summary:* DRAW test command for showing a discretisation points on edges New `"triepoints"` command has been implemented in `MeshTest` package. This command is intended to check the edge discretisation: when triangulation process is complete, it displays discretisation points on shape edges if the triangulation exists for this shape. The syntax of the command is as follows: `triepoints <ShapeName>`, where `ShapeName` is the name of an existing shape, face or edge.

Dependencies and Packaging

Development Environment

22320	*Summary:* `FreeImage` support cannot be enabled in OCCT 6.5.0 when compiling with `autotools` `Makefile` building procedure has been updated to work correctly with optional 3-rd party products (`tbb, gl2ps` and `Freeimage`)

Products

Note: Starting from this version the delivery of Products will contain the source code of the corresponding DRAW commands.

Express Mesh

22402	*Summary:* Avoid destroying discrete faces in `ComputeQuadTreeOnly` mode `QMShape_Tessellator` class has a special mode of work, when it does not create triangulation on the input faces, but rather computes only `QuadTree` decomposition leaving it on discrete faces. By mistake these faces were destroyed just after creation. This behavior has been corrected to destroy them only in the normal (creation of triangulation) mode.
22501	*Summary:* Part of face disappears in the result triangulation `QMBgr_FacetBuilder` algorithm of facet building has been corrected to avoid intersections between facet edges.

Surfaces from Scattered Points

22496	*Summary:* Restoring removed possibility in `PlateFEAPI` The possibility to set points to be approximated directly (as array) has been restored after having been lost after the implementation of issue 22110.

Collision Detection

22162	*Summary:* Collision malfunction with non-zero offset Point-in-triangle algorithm has been changed to compute barycentric coordinates, which allows avoiding false collision detection in some cases.

DXF

22363	*Summary:* Importing ArmChair1.dxf in XDE sample leads to crash Null-check has been added to `DXFCAFControl_Reader` class for translation of `INSERT` DXF entities which do not correspond to any drawable object.

Porting to version 6.5.1

Porting of user applications from the previous OCCT version (6.5) to version 6.5.1 requires the following major issues to be taken into account:

- Method Graphic3d_Structure::Groups() now returns Graphic3d_SequenceOfGroup. If this method has been used, the application code should be updated to iterate another collection type or if Graphic3d_HSetOfGroup is required to fill its own collection:

  ```
  const Graphic3d_SequenceOfGroup& aGroupsSeq = theStructure.Groups();
  Handle(Graphic3d_HSetOfGroup) aGroupSet = new Graphic3d_HSetOfGroup();
  Standard_Integer aLen = aGroupsSeq.Length();
  for (Standard_Integer aGr = 1; aGr <= aLen; ++aGr)
  {
   aGroupSet->Add (aGroupsSeq.Value (aGr));
  }
  ```
- All occurrences of `Select3D_Projector` in application code (if any) should be replaced with `Handle(Select3D_Projector)`.
- The code of inheritors of `Select3D_SensitiveEntity` should be updated if they override `::Matches()` (this is probable, if clipping planes are used).
- Constructor for `V3d_Plane` has been changed, so the extra argument should be removed if used in the application. It is necessary to add a new plane using method `V3d_Viewer::AddPlane()` if `V3d_Viewer` has been used to manage clipping planes list (this doesn't affect clipping planes representation). Please, have a look at the source code for new DRAWEXE `vclipplane` command in `ViewerTest_ObjectsCommands.cxx, VClipPlane` to see how clipping planes can be managed in the application.

If you are porting from an older version of OCCT, consult the similar section in OCCT 6.5 Release Notes document.

Supported Platforms and Pre-requisites

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

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

Linux Operating System	32/64-bit: Debian 4.0, Mandriva 2008*
Windows Operating System	32/64-bit: MS Windows SEVEN / VISTA SP2 /XP SP3
Minimum memory	512 Mb, 1 Gb recommended
Free disk space (complete installation)	650 Mb of disk space, or 1,4 Gb if installed with reference documentation
Minimum swap space	500 Mb
Video card	**GeForce** The following versions of GeForce drivers are recommended: *For Linux:* 64-bit Version: 100.14.19 or later 32-bit Version: 100.14.19 or later *For Windows:* Version 266.58 WHQL or later is recommended: http://www.nvidia.com/Download/index.aspx
Graphic library	OpenGL
C++	*For Linux:* GNU gcc 4.0. - 4.3.2. *For Windows:* Microsoft Visual Studio .NET 2005 SP1** with all security updates Microsoft Visual Studio .NET 2008 SP1 Microsoft Visual Studio .NET 2010
TCL (for testing tools)	*For Linux:* Tcltk 8.5 http://www.tcl.tk/software/tcltk/8.5.html *For Windows:* ActiveTcl 8.5 http://www.activestate.com/activetcl/downloads
Qt (for demonstration tools)	Qt 4.6.2 http://qt.nokia.com/downloads
Freetype (OCCT Text rendering)	freetype-2.3.7 http://sourceforge.net/projects/freetype/files/
Ftgl (OCCT Text rendering)	ftgl-2.1.2 http://sourceforge.net/projects/ftgl/files/
FreeImage*** (Support of common graphic formats)	FreeImage 3.14.1 http://sourceforge.net/projects/freeimage/files/Source%20Distribution/
gl2ps*** (Export of OCCT viewer contents to vector graphic file)	gl2ps-1.3.5 http://geuz.org/gl2ps/
TBB*** (Tool for parallelized version of `BRepMesh` component)	tbb30_018oss http://www.threadingbuildingblocks.org/

- * Mandriva 2008 is a permanently tested platform.
- ** The official release of OCCT for Windows contains libraries built with VC++ 2005.
- *** This product is optional.