Open CASCADE 6.3
Minor Release

Release Notes

Overview

Open CASCADE Technology 6.3 is a minor release, which includes new features, improvements and bug fixes, over minor public release 6.2. and maintenance release 6.2.1.

Version **6.3** is binary incompatible with the previous versions of Open CASCADE Technology, so applications linked against a previous version must be recompiled to run with this Version 6.3.

Table of Contents

Highlights

- Open CASCADE
  - Support of UTF8 encoding for extended strings, and Unicode symbols in IGES
  - Next step in thread-safety: protection against concurrent construction / destruction of Handle objects
  - Improved compatibility with STL and Windows-specific code
  - Multiple new features introduced in visualization module
  - New visualization library NIS (New Interactive Service)
  - New standard attributes and numerous improvements in OCAF
  - Integrated code changes made by OCC users for MacOS X and FreeBSD porting
  - Improved support of perspective view in Open CASCADE viewer
  - New version of the OCAF binary persistence format
  - The functionality of reading/writing VRML2.0 files has been implemented
  - The definitions of Standard_CString and Standard_ExtString (typedefs) have been changed to be const: from char * (or short *) to const char * (or const short *)
  - New supported platforms : Windows Vista, Mandriva2006, 2007, 2008, Debian Etch, Red Hat Enterprise 4.0
  - New supported compiler: gcc 4.0-4.2, Visual C++ 8.0
- Products
  - Reading of ACIS interface SAT versions 10.0 - 14.0
  - Improved tools for reading and writing NASTRAN files in OMF

New Features

Foundation Classes

News in version 6.2.1 over 6.2

- Support of UTF8 encoding has been implemented into `TCollection_ExtendedString.`

  A string in UTF8 encoding is supposed as pure `CString` terminated by null.

  The constructor `TCollection_ExtendedString (const Standard_CString astring, const Standard_Boolean isMultiByte)` now has an additional parameter `<isMultiByte>`, by default "false". If it is "true", the input `CString` is supposed to be in UTF8 encoding.

  Two methods have been added:

  - `LengthOfCString()` returns Integer.
    Returns the expected `CString` length in UTF8 encoding. It can be used for memory calculation before converting to `CString` in UTF8 encoding.

  - `ToUTF8CString(theCString : out CString from Standard)` returns Integer;
    Converts the `string` to UTF8 encoding and returns length of the resulting CString. A memory for the `<theCString>` should be allocated before call, with the size computed using the previous method `LengthOfCString()`.
- New method `Added()` is provided in the class `NCollection_Map`. This method adds new key to the map (unless the same key is already there) and returns const reference to the corresponding key actually contained in the map. Thus it provides access to the object instances contained in the map, which can be useful when the map is used as a means to ensure uniqueness (by value) of instances of objects of the definite type.
- Two new template classes have been added into the `NCollection` package:

  - `NCollection_SparseArray:` provides a compact data structure optimal for storing small objects addressed by integer index, where not full range of indices may be occupied. This data structure is similar to the map with integer keys but is more compact and provides better performance.

  - `NCollection_CellFilter:` provides a tool for fast check of coincidence or overlapping of geometrical objects (e.g. points)
- New methods have been added to classes `Bnd_B2x` and `Bnd_B3x` (2- and 3-dimensional bounding box):

  - `IsVoid()` indicates whether the box is uninitialized. The uninitialized state should be checked before calling `CornerMin(), CornerMax()` and `SquareExtent()`, otherwise these methods return irrelevant results;

  - `IsIn (theBox)` checks if the box is completely inside the given box.

  - IsIn (theBox, theTrsf) checks if the box is completely inside the given box transformed with the given transformation.

  - The 3rd parameter defining if the circle/sphere is hollow has been added into the method `IsOut()` checking for intersection of the box with a circle/sphere,. If the parameter is "true", a box that is completely inside the circle/sphere is not considered in intersection.

  - The 3rd parameter `theOverthickness` has been added into the method `Bnd_B3x::IsOut (theLine, isRay)`. It defines the addition to the size of the tested box; it can also be understood as the radius of the cylinder that is checked for intersection instead of the given line.

- New static functions `IsReentrant()` and `SetReentrant()` have been added into the Standard package to query and set flag for reentrant (i.e. thread-safe) mode of operation of `TKernel` services (memory manager, handles, exceptions) in run-time. Note that the default value of this flag is True if environment variable MMGT_REENTRANT is defined and False otherwise.
- New class `Message_Algorithm` has been added, intended to be a base class for algorithms. It provides descendant classes with uniform means to give feedback on the execution of the algorithm by setting status flags and print messages, configurable via resource file.

  A few related types have been added:

  - class `Message_ExecStatus` represents a cumulative status of the algorithm that can be a composition of several status flags
  - enumeration `Message_StatusType` defines available types of status flags `(Done, Warning, Alarm, or Fail)`
  - enumeration `Message_Status` defines 32 status flags that can be set for each status type

News in version 6.3 over 6.2.1

- A new class `Poly_CoherentTriangulation` (and the types on which it depends) has been added to the package `Poly` (toolkit `TKMath`).

  This class implements the data structure and the API that support analysis and modification of the triangulated mesh (like `Poly_Triangulation`). This tool can be used in the following areas:

  - Mesh generation and refinement
  - Mesh decimation
  - Constructions on surface meshes: offset, free boundary, geodesic, etc.

  The data structure has been optimized to provide the best performance of the algorithms.

  The complete Doxygen documentation is contained in `Poly_CoherentTriangulation.hxx`

- Possibility to store vectors of normal to the surface in vertices of the triangulation structure (class `Poly_Triangulation`) has been introduced. This provides better performance of visualization (recalculation of normal is avoided) for normal shapes and allows visualization of shapes having only triangulated representation (without any other geometry - curves, surfaces ...) with smooth shading.

Modeling Algorithms

News in version 6.2.1 over 6.2

- New functionality provides shape volume computation by adaptive Gauss-Kronrod method. New computation methods are accessible via new methods of class `BRepGProp: VolumePropertiesGK(...)`.
- New method `BRepAlgoAPI_BooleanOperation::RefineEdges()` is added in class BRepAlgoAPI_BooleanOperation.
  This method can be used after performing any Boolean operation to concatenate edges that belong to the same face, have at least G1 continuity in common vertices and the same type of underlined curves (except Offset curves).

News in version 6.3 over 6.2.1

- New parameter "Toler" which defines tolerance when calculating curve length has been added to class `GCPnts_UniformAbscissa`. By default this parameter is equal to `Precision::Confusion()`.
  It is required to use with this class the same tolerance as in this algorithm for curve length calculation. Otherwise results could be inconsistent.
- Open CASCADE tessellation algorithms (used for building triangulation of shapes in visualization) have been unified and improved. The improvements include protection against duplicated 2D nodes, elimination of numerical instabilities, advanced handling of holes and cutting of external triangles, optimization of `MeshDS_Mesh2d` data structures and performance of remeshing in self intersection case.

  A new possibility is provided to customize tessellation algorithms used in OCCT visualization via plug-in mechanism. The basic classes for plugins are:
  - `BRepMesh_PluginMacro`
  - `BRepMesh_DiscretFactory`
  - `BRepMesh_DiscretRoot`
- Progress indication for reading and saving of BRep files has been implemented. For that, progress indicator can be passed as an additional optional argument to methods BRepTools::Read() and BRepTools::Write(). The applications can feature the progress bar with the possibility to stop the indicated operation.

  Note that it is recommended to update the bar not more than 10 times in 1 second to avoid performance decrease.

Visualization

News in version 6.2.1 over 6.2

- Selection mechanism in `MeshVS` has been improved to support a great number of selectable mesh entities (500K and more) with a good response time. This new functionality has been introduced as an option. According to the new approach, the single selectable object is created for all presentation, and the treatment of the selection event is exercised on application level.

  The old approach remains the default one. It consists in creating in memory a separate object for every selectable item-owner. This approach becomes inefficient when the number of objects exceeds 100 000.

  Several new virtual methods of `MeshVS_DataSource` are to be redefined to switch to the new mechanism. The virtual method `IsAdvancedSelectionEnabled()` defines if the new mechanism is used.
- Simplified visualization of vectors has been implemented in `MeshVS` package. The simplified visualization mode can be activated instead of the present vector presentation through a special flag and parameters (see methods `SetSimplePrsMode()` and `SetSimplePrsParams()` of the class `MeshVS_VectorPrsBuilder`). It allows faster computation and less memory allocation for presentations on large mesh models.
- The possibility to read depth values of OpenGL window pixels has been implemented for applications that use OCC viewer. The class `Visual3d_View` provides the necessary API for that (see method `ReadDepths()`).
- Support of mesh group selection has been introduced in `MeshVS` package. To enable this functionality group selection mode should be activated using new `MeshVS_SelectionModeFlags` enumeration, which allows defining a specific mesh selection mode:

- MeshVS_SMF_Mesh = 0 – selection of the whole mesh (default)
- MeshVS_SMF_Node = 1 – selection of mesh nodes
- MeshVS_SMF_0D = 2 – selection of 0D elements
- MeshVS_SMF_Link = 4 – selection of edges
- MeshVS_SMF_Face = 8 – selection of faces
- MeshVS_SMF_Volume = 16 – selection of volumes
- MeshVS_SMF_Group = 256 – selection of mesh groups

Information about the existing groups is provided by MeshVS_DataSource with the following methods:

- GetAllGroups() returns list of all defined groups (group IDs);
- GetGroup() returns type and list of elements of given group;
- GetGroupAddr() returns address to group data structure if defined. The obtained address value is used for selection to provide quick access to group info from the selected owner.

Note that the default MeshVS_DataSource class implementation returns no group information. The real data should be provided by MeshVS_DataSource class successor.

Additionally the following modifications have been implemented:

- ComputeSelection() method from MeshVS_Mesh have been modified to build a set of sensitive entities for each group (process MeshVS_SMF_Group selection mode). A sensitive entity is built only for selectable elements/nodes belonging to a given group. If the group contains no selectable elements/nodes, then a dummy sensitive entity is created.
- HilightSelected() and HilightOwnerWithColor() have been modified to provide highlighting of detected group.
- MeshVS/MeshVS_MeshPrsBuilder::BuildElements() method have been modified to support highlighting of a set of elements (process MeshVS_DMF_HilightPrs display mode).
- MeshVS/MeshVS_Owner.cdl and MeshVS/MeshVS_Owner.cxx contain a new flag myIsGroup indicating the owner of a group of elements/nodes (default value is false). The flag value can be asked with a new IsGroup() method. In the case of a group owner (IsGroup() returns true) the Owner() method returns the address of the group data structure that corresponds to the value returned by MeshVS_DataSource::GetGroupAddr() method.

- The Boolean parameter ContainsFacet (false by default) has been added to Graphic3d_Group::UserDraw() method, indicating whether the primitive contains filled polygons and therefore requires the depth buffer. If the parameter is true, the function updates its MyContainsFacet flag.
- Possibility to set the transparency for the objects of Visual3d_Layer has been implemented.

News in version 6.3 over 6.2.1

- The data structures and visualization routines supporting visualization of voxel models have been added in the new package Voxel. A voxel is a sub-volume box with attached constant scalar/vector value. The object in voxel representation is split into many small sub-volumes (voxels) and its properties are distributed through voxels.

  In the current implementation OCCT provides several basic data containers of voxels with fast access to the data and optimized allocation of data in memory.

In the same time it provides simple Boolean operations (fuse and cut) on cubes of voxels and a mechanism of voxelization conversion of a geometrical model into a voxel representation (discrete topology).

A specialized classes allow visualization of voxels as colored or black/white points and cubes, displaying only the voxels visible from the user's point of view.

- The possibility to record an AVI video stream from OpenGl code has been implemented. The class `OpenGl_AVIWriter` provides the API to create an AVIfile, start/stop the recording and to close the file. It uses Windows Multimedia API for this purpose, thus the code can only be executed on Windows platform.

  The new DRAW command "vrecord" can be used to test the AVI creation in 3D AIS viewer (ViewerTest package). This command also allows choosing the AVI codec (by default it uses XVID).
- A special class `V3d_LayerMgr` has been introduced in OCCT to control the layer update taking into account various 'macro' objects that must be rendered simultaneously (such as labels, color scale etc.).
- The new visualization library 'NIS' has been created, providing the API similar to the traditional AIS but with some important differences/improvements:
  - For every type of InteractiveObject a corresponding Drawer class should be provided that defines the presentation of the Object type using direct OpenGl calls. This is a much faster way to display 3D objects, allowing to have more than 1 million separate selectable entities in one view.
  - The abstract type NIS_InteractiveObject does not support any properties (color, material, other aspects). The relevant properties should be defined in the specializations of the Drawer class, and the API to set/modify should be implemented in the specializations of InteractiveObject class.
  - Interactive selection is managed by InteractiveObject methods instead of special selector classes and data types. This is possible since in NIS the selection is based on 3D representation (by a ray or a box corresponding to the view direction) without intermediate 2D projection.
  - Many InteractiveContext instances can be attached to a V3d_View, these instances being independent containers of interactive objects; removal (detaching) of InteractiveContext instance destroys the contained objects.
  - All data types and algorithms are designed to provide the best performance for both OpenGl (server side) and application. On the other hand, the API is open to any feature supported by any version of OpenGl. This allows to build custom presentations quickly and efficiently.
  - The type NIS_View subclasses V3d_View thus providing all its public API like scene definition (view orientation, lights, background, etc.) as well as the standard view transformations (pan/zoom/rotate,fitAll,...). The traditional AIS-based presentations (e.g., AIS_Shape) are also supported, they can be rendered together with NIS presentations in the same view window.
  - The DRAW test plugin, TKViewerTest, is modified so that it manages now AIS_InteractiveContext and NIS_InteractiveContext together in one view window.
- The new `ErasedObjects` command gives access to objects with status `AIS_FullErased` from context. These are the objects which have been erased from display but not placed into the interactive context collector. Previously they were not accessible, which could lead to memory leaks.

  The new ObjectsByDisplayStatus command gives access to objects from context by display status (`AIS_Displayed, AIS_Erased, AIS_FullErased`).

Application Framework

News in version 6.2.1 over 6.2

- Boolean flag `<isCheckItems>` has been added to the methods `ChangeArray()` of attributes `TDataStd_IntegerArray, TDataStd_RealArray` and `TDataStd_ExtStringArray`. This flag indicates whether comparison of each item of the input array with the items of the internal array of OCAF attribute is necessary.
    - If the flag is "true" (default value), the OCAF attribute compares the arrays and calls the backup methods only if the arrays are different (i.e. the user actually changes the array).
    - If the flag is "false", the input array replaces the internal array of OCAF attribute without any check. The flag is relevant when dimensions of arrays are equal.

  Additionally, the methods: Value(), Upper(), Lower(), Length(), and Dump() have been optimized.

- A number of new data attributes have been added in the TDataStd package.
    - `Tick` – defines a boolean flag attached to a label.
    - `AsciiString` – defines a pure ASCII string.
    - `IntPackedMap` – defines a map of integers packed so that it occupies minimum space in memory.

  Lists:
    - `IntegerList` – defines a list of integer values.
    - `RealList` – defines a list of double values.
    - `ExtStringList` – defines a list of extended strings.
    - `BooleanList` – defines a list of boolean flags.
    - `ReferenceList` – defines a list of references to labels.

  Arrays:
    - `BooleanArray` – defines an array of boolean flags.
    - `ReferenceArray` – defines an array of references to labels.
    - `ByteArray` – defines an array of unsigned chars.
    - `NamedData` – keeps named data in internal maps, i.e. it keeps pairs Key – Value, where a Key – `ExtendedString` and a Value belongs to one of the following types: `Integer, Real, ExtendedString, Byte, ArrayOfIntegers, ArrayOfReals`.

News in version 6.3 over 6.2.1

- New package `TObj` has been implemented as a separate `TKTObj` toolkit. The purpose of this package is to facilitate implementation of object-oriented data model on top of OCAF data structures. This package provides data objects serving as intermediate layer between OCAF data attributes and high-level application code.

  This set of classes contains specific attributes and implementation of base classes for object, model and application, with support of Undo/Redo and persistence (currently only binary and Xml).

  All objects store their data (simple data, child objects, references) as attributes in OCAF document on their sub-labels. The advantage is that the created object-oriented layer allows working with objects and their data and relationship without dealing much with internal OCAF structures and without additional implementation efforts of persistence for each new created object type.

  The set of classes covers such basic concepts as application, model, object, references between objects, iterators, etc.

- A new mode of storing transaction delta has been implemented in some OCAF attributes. It is possible to use these attributes both in the standard mode, when the full attribute value is backed up during `CommitTransaction` command, and in new `DeltaOnModification` mode, when only the difference between the new and the old attribute value is backed up. The new feature can be used when the attribute value (array or packed map) of a very great size is expected and the usage of standard backup mode becomes not effective.

The implementation allows using the same attributes in the same document both in the standard mode and in `DeltaOnModification` mode. The choice is driven by Boolean parameter `<isDelta>` of the corresponding `<Set>` command of the attribute. It is required that the mode (standard or `DeltaOnModification`) of attribute usage will not be changed after the initial set. If it is necessary for the application logic to change the mode for existing attribute object, that attribute should be removed using `Forget` command and created anew.

- Function Mechanism of Open CASCADE Application Framework (OCAF) has been extended with the following possibilities:
  - Direct manipulation of arguments and results by a function
  - Creation of a graph of functions by means of dependencies between functions determined by their arguments and results
  - Iteration of the graph of functions in execution order
  - Class – interface for Function Mechanism
  - Support of multi-threaded mode of execution of the functions (calculation of the maximum number of threads, a list of functions for being executed simultaneously)

- New version of the OCAF binary persistence format has been implemented.

  This improvement unifies the inclusion of non-OCAF data into persistent documents. In format version 2 there was only one such data block in Shapes section. In the new format version 3 it is possible to create any number of custom sections, the Shapes being just one of them.

  Each Section is uniquely identified by ASCII name string and also has the attribute `IsPostRead`. This attribute defines if the Section is read before or after the OCAF data. Earlier the BinOcaf attribute drivers could access non-OCAF data read in advance (as in the SHAPES section).

  To use custom sections in OCAF binary documents, it will be necessary to subclass `DocumentRetrievalDriver` and `DocumentStorageDriver` from BinLDrivers, with the relevant definition of virtual methods that process the section data.

  Compatibility with documents written in format version 2 is preserved in `DocumentRetrievalDriver.`

Data Exchange

News in version 6.3 over 6.2.1

- The functionality of reading/writing VRML2.0 files has been implemented in OCC in a new package VrmlData. The API to read and write VRML files is provided in the class VrmlData_Scene. Note that this functionality is completely independent on previously existing VRML 1.0 writer (API provided by the package VrmlAPI), due to principal differences between the two formats.

Draw

News in version 6.2.1 over 6.2

- New function "countshapes", which returns number of shapes taking into account their orientations and locations has been introduced.
- Two commands have been added to OCAF Draw module:
  - `SetUTFName DF, entry, fileName` allows importing a UTF8 text from file `<fileName>` and storing it in the current document under the tree of sub-labels. The label `<entry>` is a parent label for it.
  - `GetUTF DF, entry, fileName` allows extracting text information (strings) stored in the specified document under the tree of sub-labels. All strings found under child sub-labels of <entry> are converted in UTF8 and are kept in the specified file <fileName>.

Products

News in version 6.2.1 over 6.2

OMF

- A new low-level class `OMFTools_NASFormatter` has been added in the package `OMFTools` for writing data according to NASTRAN requirements for formatting of different types of data (integer, double, string, comment).

New supported platforms and compilers

- Open CASCADE has been ported to the following new platforms:
  - Windows Vista,
  - Mandriva2006, 2007, 2008,
  - Debian Etch,
  - Red Hat Enterprise 4.0
  - compiler gcc 4.0-4.2,
  - Visual C++ 8.0

Improvements

Foundation Classes

Improvements in version 6.2.1 over 6.2

- Reference counter in `Standard_Transient` class (and consequently in all OCCT classes managed by handles) is protected to be thread-safe, with respect to simultaneous construction and destruction of Handles to the same object.

  Note that this feature is optional; it should be activated when needed either by defining environment variable MMGT_REENTRANT before starting the OCCT application, or by calling static function `Standard::SetReentrant(Standard_True)`.

  The current implementation of protection of the reference counter has some limitations.

  - The protection now works only on Windows and Linux platforms
  - This improvement leads to some downgrade of performance of creation and destruction of Handle objects (~2 times when reentrant mode is OFF, ~10 times when it is ON). So, it becomes more important to avoid creation of unnecessary Handles in performance-critical parts of the code. For information, the time of creation and destruction of one Handle measured on P4 3GHz CPU is around 20 ns in non-reentrant mode and 100 ns in reentrant mode.
- The bug in method `Subtract` of the class `TColStd_PackedMapOfInteger` that led to inconsistency of the map has been corrected.
- The problem with running out of memory that might cause mutex deadlock in multithreaded applications running under UNIX/Linux has been fixed
- The numerical problem in constructor `gp_Lin2d` has been fixed. Previously, the origin of the created line was located at the intersection of the line with axis OX. With small "`A`" this brought about a very large X-coordinate of the origin point.

  The corrected constructor of class `gp_Lin2d` creates the line origin as the nearest point to the global origin (0, 0) located on the line. This algorithm is simpler than the previous one that created the origin on axis OX or OY, and it avoids any numerical problems whatever the parameters of this constructor.
- Implementation of type information in OCCT has been corrected for method `Standard_Type::Size()` to return the actual size of the corresponding type. Previously the value returned by this method was actually a length of the type name.
- Method `Bnd_Box::Add(Bnd_Box&)` has been corrected to avoid taking the value of the gap into account twice.
- A set of changes in OCCT code and WOK configuration files made by OCCT users to ensure its porting on other platforms (MacOS X – by Torsten Sadowski and GMSH, FreeBSD – by Thomas Thierry) have been integrated. This does not make OCCT completely ported on these platforms, but facilitates such porting in the future.
- Granularity of memory manager allocation blocks on SGI64 bit has been increased up to 8 bytes instead of usual 4 bytes on 32 bit systems.

Improvements in version 6.3 over 6.2.1

- Possibilities to have different cell size on each dimension and to remove objects from the structure have been introduced in the template class `NCollection_CellFilter`.
- The bug in the method `TColStd_PackedMapOfInteger::Subtract()` has been fixed. The problem was that in some cases (typically for a very great number of integers in the map) call to

method Subtract() could lead to loss of some numbers stored in it (with Extent() returning correct value, but iterator skipping some values).

- Copy-constructor of class math_IntegerVector has been moved from private to public block.
- Some OCCT header files have been corrected to avoid compiler warnings when high warning level is set (e.g. level 4 on MS VC++ compiler).
- NCollection containers (maps, list and sequence) have been improved to use allocators, allowing to reduce the number of calls to the system memory routines:
  - Map collections now can accept an allocator in their constructors. In this case, the possibility to clear a map without releasing the memory used for the table of buckets has been implemented to effectively reuse the same map many times.
  - The methods operator=() in List and Sequence collections now replace their own allocator with the allocator of the other collection. This allows creating collections where List or Sequence is used as an item type, using the same allocator for all structures.

Modeling Algorithms

Improvements in version 6.2.1 over 6.2

- The algorithm of 2D curve building used by ShapeFix_Edge class has been fixed to avoid creation of the curve if the first and the last point coordinates are equal.

Improvements in version 6.3 over 6.2.1

- The algorithm of calculation of section between 2 tori has been improved for the cases when two tori have the same radiuses and center, but different axes.
- Bounding box is now calculated precisely for case of analytical sphere.
- Some memory leeks in BOP have been eliminated.
- To improve the performance of BRepTools, parametric curves are from face's edges only once and stored in intermediate lists.
- The way of treatment of intersection between two faces that have the same underlying surfaces has been changed. Now the case is treated by the common general way that is used for any pairs of faces.
- The treatment of seam edges that belong to faces with B-Spline and Bezier underlying surfaces has been improved.
- Treatment of a hyperbola has been changed to prevent infinite result during computation of hyperbolic sine.
- ReShape function has been modified to keep the natural restriction flag during face to face modification. This flag should be kept if the old face is transformed into one new face (1 to 1 modification). Please note that if face is split (1 to N modification) this flag is not kept.
- The class ShapeUpgrade_SplitCurve3dContinuity has been corrected to avoid the exception in case of complete knot removing.

Visualization

Improvements in version 6.2.1 over 6.2

- The bug in Bnd_BoundSortBox2d that sometimes prevented selection of objects by mouse, for example, when all selectable objects were fit along a horizontal line of the view port, has been fixed.
- Anti-aliasing visualization has been corrected to work both in normal and environment mapping modes.

- The bug in the method `V3d_View::Convert(theX, TheY, aXpix, aYpix)`, which generated incorrect output pixel coordinates when “top” or “side” orientation has been set in the view, has been corrected.
- The bug with exclusion of unprocessed elements in `MeshVS_PrsBuilder` has been corrected. Now the builder excludes an element only if it really has been processed.
- The bug causing exception during visualization of empty nested compounds has been fixed.
- Calls to the function `exportText` have been eliminated in `OpenGl_tXfm.c`, unless OpenGL is in feedback mode.
- The class `DsgPrs_DiameterPresentation` has been modified to increase the space between the symbol of the diameter and the diameter value, which earlier was partially hidden by the symbol.
- Instruction `"C++: alias ~"` for the method `Destroy()` in `WNT_PixMap.cdl` (lost in OCCT 5.2.3 release) has been restored, thus fixing the problem of destruction of instances of this class.
- A detailed comment for method `SetTransparency()` has been added in `Graphic3d_MaterialAspect.cdl`.
- The bug in `AIS_DiameterDimension` raising exception if the automatic-position of the dimension is OFF has been fixed.

Improvements in version 6.3 over 6.2.1

- The bug in distribution of memory in C layer of visualization library has been fixed. Earlier, in case of many interactive objects (more than 22 000), the memory became so fragmented that at least 100kb could not be allocated anymore.
- Initialization of `Select3D_Box2d` class used by interactive selection has been corrected so that the box coordinates were updated properly and thus to picking errors would have been avoided.
- Hidden element IDs are excluded from iterations and from memory allocation during mesh presentation creation. A default constructor has been added for class `MeshVS_TwoNodes`
- Uninitialized variables in `MeshVS_Tool` are now initialized with the default status.
- Revision of perspective view implementation in Open CASCADE viewer has been started.

  The following convenient methods have been added to V3d API in order to facilitate definition of perspective view parameters:

  - `V3d_View::DepthFitAll(Aspect,Margin)`: adjusts Z size and depth of a 3D view automatically so as not to clip the displayed objects by front and back clipping planes. Depth is Aspect / 2 times smaller than the new Z size, that allows achieving required perspective effect easily.
  - `V3d_View::SetViewingVolume(Left,Right,Bottom,Top,ZNear,ZFar)`: allows defining viewing volume parameters in OpenGL-like manner (see `glFrustum()` and `glOrtho()` functions). It computes view orientation and mapping parameters on a basis of its arguments, depending on the view type (orthographic or perspective).
  - `V3d_PerspectiveView::SetPerspective(Angle,UVRatio,ZNear,ZFar)`: convenient function that works similarly to `gluPerspective()`. It calculates view orientation and mapping parameters on a basis of its arguments.

  Low-level part of OCCT visualization that concerns working with view orientation (MODELVIEW) and mapping (PROJECTION) matrices has been simplified. These matrices are calculated in a single source file (OpenGl_view.c) and re-used by OpenGL visualization, interactive detection and selection.

  In further OCCT versions, this simplification and refinement will be continued in part concerning auxiliary algorithms such as "fit all" operation and the code that projects 3D points onto the viewport.

- OCCT Visualization User's Guide has been amended with a detailed description of view definition parameters. The recommended parameter values suitable in most typical cases are given.

Application Framework

Improvements in version 6.2.1 over 6.2

- The bug with XML OCAF reader considering "00", "000", etc. as integers and therefore reading them as "0", which caused problems with the use of names like "00" in `TDataStd_Name` attribute has been fixed. The name "0000" is now correctly read by OCAF XML reader as "0000".
- The method `LDOM_XmlReader::ReadRecord(LDOM_OSStream& theData)` has been corrected. If the attempt to read record is not successful (return code = -1), then the processing of the current record is canceled and the method returns XML_UNKNOWN type of record. The <myError> receives message "Unknown read/write error".
- In operations of saving and opening OCAF document it has become possible to use local file names (without directory prefix). If a directory name is not set it is assumed that the current directory will be used.
- The signatures of some methods in `TDataStd` package that return `TCollection_AsciiString` and `TCollection_ExtendedString` have been corrected to return a reference instead of a copy of the string, thus avoiding unnecessary operations and improving the performance.
- Storage of information about vertexes in the format of the binary persistence file has been modified. The new format correctly processes geometry having points on Curve or point on Surface or points on curve of surface. The old format is also interpreted correctly.
- The exception raising if `TDF_Label Resume` operation was executed with a removed attribute has been fixed.
- In the previous version, a reference to an empty label became lost after opening of the document. Now the document keeps the reference.
- Now the user gets a proper warning of the failure if document is being saved to a read only file or directory.

Improvements in version 6.3 over 6.2.1

- The regression connected with impossibility to change the range of the arrays already attached to a label has been fixed.
- The performance of `Append` operation has been optimized.

Data Exchange

Improvements in version 6.2.1 over 6.2

- The methods for writing shape in the file for STEP, IGES and BREP formats have been improved to check the state of stream after saving the data on disk. If the file was not saved correctly (for example the disk is full), the following methods will return the error status.
  - `IGESControl_Writer::Write(CString,Boolean)` returns `Standard_False`;
  - `STEPControl_Writer::Write(CString)` returns `IFSElect_RetError`;
  - `BRepTools::Write(TopoDS_Shape,CString)` returns `Standard_False`;
- Earlier OCC IGES reader incorrectly handled Unicode data from IGES header. If the code of the symbol was more than 127 the processing of header was incorrect (which caused, for example, wrong reading of units). This fix corrects the reading of header (useful for non-European, for example Japanese vendors).

- A bug has been fixed in `Interface_ParamSet::Append()`: due to incorrect allocation of memory buffer, reading IGES files containing long text strings in the General section could be a cause of memory corruption.
- The algorithm of IGES assembly translation has been fixed to work with non-mm units.
- `IGESData_IGESEntity` class has been fixed to handle correctly several names of one IGES entity in an input file. `NameValue()` method returns the first available name, while `TypedProperty()` method allows getting the n-th value of a property in general.
- Reading of Ordered Groups (Type 402 Form 14 and 15) from IGES file format has been implemented.

Improvements in version 6.3 over 6.2.1

- The implementation of class `Interface_ParamList` has been changed so that now the parameters are stored in class `NCollection_Vector` instead of class `TCollection_HArray1` when a file is loaded.
- The protection on reading validation properties without the name has been implemented.

Building tools

Makefiles procedure

- **make install** step installs Open CASCADE in usable form without any additional operations.
- **`make install`** step of Makefile procedure has been modified to correctly copy OCCT sources.
- **configure** script of OCCT has been improved to automatically detect 64-bit platforms.
- Problem with permissions during re-launch of make install of the Makefile procedure has been eliminated.
- New flags have been added to OCCT Makefile procedure. Now you can build Open CASCADE without Draw, WOK and Jcas modules.

  Additional flags:

  - --disable-draw – allows OCCT building without Draw.
  - --disable-wok – allows OCCT building without WOK.
  - --disable-wrappers – allows OCCT building without JCas (Wrappers).

  If you want to build Draw, Wok or Jcas, please, define the following flags:

  - Draw - `--enable-draw=yes –with-tcl=`/path-where-tcl-is-installed `–with-tk=`/path-where-tk-is-installed
  - Wok - `--enable-wok=yes –with-tcl=`/path-where-tcl-is-installed `–with-tk=`/path-where-tk-is-installed
  - Jcas - `--enable-wrappers =yes –with-java-include=`/path-where-java-includes-are-installed

MS Visual Studio projects

Files for MS VC++ 6.0, VC++ 7.1 and VC++ 8.0 have been placed in the directory **ros/adm/win32** in the folders vc6, vc7 and vc8 correspondingly. It is possible to use them to rebuild OCCT using either VC++ 6.0, VC++ 7.1 or VC++ 8.0.

New **ros/adm/win64** folder contains **vc8** folder with MS Visual Studio project files for MS VC++ 8.0 that you can use to rebuild Open CASCADE in 64 bit mode.

3rd party products

Qt 3.3.3 has been replaced by Qt 4.2.3 in 3rdparty folder of Open CACADE delivery.

Samples and Tools

Open CASCADE QT Import Export sample, tutorial, graphic3ddemo and OCAF browser have been ported to Qt 4.2.3.

Warnings

The first step of eliminating Open CASCADE compilation warnings was finished.

Open CASCADE compilation gives no warning on Linux platform by Makefiles building procedure without defining the level of warning output (i.e. by default)

Products

Improvements in version 6.2.1 over 6.2

OMF

- NASTRAN reader has been improved and is now able to handle properly different types of continuation entries.
- Class OMFTools_NASFormatter has been completely updated following recent changes in OCCT (const in Standard_CString).

Parasolid interface

- Reading of the modern versions (10.0 – 14.0) of ACIS interface SAT format has been implemented.

ACIS interface SAT

- Processing of ACIS interface SAT topological entities possessing tolerance `(tedge, tvertex and tcoedge)` has been improved. Earlier they were not taken into account.
- The ACIS interface Reader inability to read ext fields containing # symbol has been fixed: the separator is handled only if it is a separate field.
- The handling of faces which lack natural boundaries has been implemented: a natural bound is added if all wires are reversed.

Collision Detection

- A method to check intersection of tested shape with a ray has been added.
- A method to check intersection of tested shape with a sphere has been added.
- The intersector has been made thread-safe (the same instance called from 2 threads).

Improvements in version 6.3 over 6.2.1

Express Mesh

- The bug in the algorithm causing some faces with holes to be incorrectly meshed (some holes were filled) has been fixed.

Changes

Foundation Classes

Changes in version 6.2.1 over 6.2

- Some changes have been made in basic OCCT headers:
  - Statement "using std" in Standard_Stream.hxx has been replaced by individual "using" statements for several types used extensively in OCCT code. This should reduce the risk of conflicts between OCCT headers and application codes due to imported STL names. However, some code probably might need to be corrected to add std:: prefix before STL classes (or equivalent "using" statement).
  - A number of macros (such as WIN32_LEAN_AND_MEAN) have been defined in Standard_Macro.hxx for WNT platform in order to reduce the number of symbols defined by eventual inclusion of windows.h. This is necessary to prevent interference between multiple macro definitions made in windows.h (e.g. SendMessage defined as SendMessageA) and normal code of OCCT and applications. If some code needs the complete set of symbols defined in windows.h, it is recommended to include windows.h prior to OCCT headers.
- Methods RowLength() and ColLength() of the class NCollection_Array2 have been corrected to return length of rows (i.e. number of columns) and length of columns (i.e. number of rows), respectively, in order to have the same meaning as the same methods in the class TCollection_Array2. Note that previously these methods had swapped meaning.
- A number of rarely-used classes have been removed from TcolStd (instantiations of Slist and AVLSearchTree templates)
- Class Message_Messenger providing a top-level interface to the messaging system has been added. This class provides the possibility to direct messages into one or more contexts, each being a descendant of Message_Printer class, and intended to be used instead of separate instances of Message_Printer classes.

  Static global instance of the messenger returned by Message::DefaultMessenger() is provided for use from parts of code that do not have other interfaces to the user. By default, this messenger contains a single context connected to cout.

  The following modifications have been implemented in existing classes:

  - Argument tracelevel of Message_Printer class methods that was an integer value has become enumeration Message_Gravity.
  - Class Message_PrinterOStream replaces previous implementation of Message_Printer and Mesage_TraceFile.
  - Removed classes: Message_OutFile, Message_TraceFile (use Message_PrinterOStream instead).
  - Removed aliases: MoniTool_Msg, MoniTool_MsgFile, MoniTool_TraceFile, Interface_OutFile, Interface_TraceFile (use corresponding classes from package Message)
  - API of Message_Msg class changed: Methods AddInteger and AddString replaced by method Arg (aliased to "operator <<")

Changes in version 6.3 over 6.2.1

- The definitions of Standard_CString and Standard_ExtString (typedefs) have been changed to be const: from char * (or short *) to const char * (or const short *). New types Standard_PCharacter and Standard_PExtCharacter are provided for cases when non-const pointers are needed. This change provides the following advantages:

  - It has become possible to pass both (const char *) and (char *) types as arguments to OCCT methods. Earlier it was only possible to pass (char *).
  - The code security has been improved. Passing a char * is unsafe: such strings generally should not be modifiable outside the scope where they are created and destroyed (exceptions are well-known functions like strcpy, strcat). To pass a modifiable string use either TCollection_AsciiString or Standard_Address type, or use directly the type (char *).

- Conversion of big double value to integer on Windows under MS VC++ 8.0 raised a signal not handled previously by OSD::SetSignal(). This has been fixed by implementation of the following changes:
  - In OSD: handling of signal "Floating point multiple traps" and converting it to appropriate OCC exception (Standard_NumericError)
  - In Standard: function RealToInt() changed to perform safe conversion to integer (using method (a), i.e. cutting values to be in the range from INT_MIN to INT_MAX).
  - In V3d: function RealToInt() is used to avoid exceptions in conversion of values

Visualization

Changes in version 6.2.1 over 6.2

- Along with implementation of the advanced selection mechanism in MeshVS package, the interfaces of some classes have been changed to increase the performance when dealing with a great number of mesh elements and to support the new advanced selection mode:
  - The class MeshVS_MeshOwner has been renamed to MeshVS_MeshEntityOwner. Note that the class with the name MeshVS_MeshOwner also exists; it represents the whole mesh as a selectable entity.
  - The class MeshVS_DataMapOfIntegerMeshOwner has been renamed to MeshVS_DataMapOfIntegerMeshEntityOwner.
  - The interface of pure virtual method GetNodesByElement of MeshVS_DataSource has been changed to avoid creation of a new array on each call.
- The interface of the methods Build and CustomBuild of the class MeshVS_PrsBuilder has been revised to avoid copying of all IDs of the whole mesh into an artificial temporary data structure MeshVS_Array1OfIntegerBoolean.

  The classes MeshVS_IntegerBoolean and MeshVS_Array1OfIntegerBoolean and the static method MeshVS_Mesh::TColStdMapToMeshVSArray have been removed.
- The small change in font management for text display on UNIX platform has been introduced. Now if the given font is not found in the font base, the default font is used (the first font in the list of font names of OCCT)

Changes in version 6.3 over 6.2.1

- The behavior of material transparency has been revised; current behavior is described in detail in Graphic3d_MaterialAspect.cdl. The dependency of the current transparency on diffuse color of the previous material has been eliminated.

Application Framework

Changes in version 6.3 over 6.2.1

- The class `TPrsStd_NamedShapeDriver.cxx` has been modified: the method `TNaming_Tool::CurrentShape` (NS) has been replaced by the `TNaming_Tool::GetShape` (NS).

  The drawback of the old method is that it returned not the shape contained in the specified NS, but the last modification of this shape. Now the method `TNaming_Tool::GetShape` (NS) returns exactly the shape value of the specified NS.

Data Exchange

Changes in version 6.2.1 over 6.2

- Data Exchange components (IGES, STEP) now use `Message_Messenger` class for outputting messages (instead of cout or `Message_Printer`). This may affect output of messages by these components. Several headers have been removed (`Interface_DT.hxx, Interface_TraceFile.hxx` etc.) along with relevant aliases and macros. The facilities provided by the package `Message` (see above) shall be used instead.

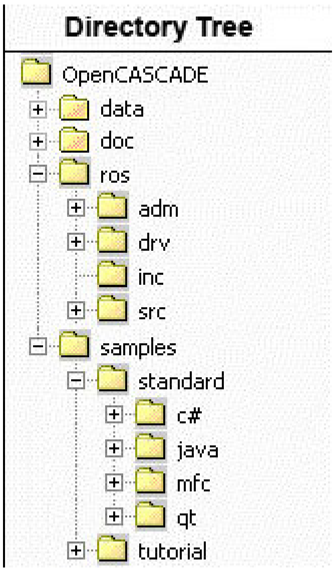
Building tools

Starting from Open CASCADE 6.3 support and delivery of COMP (UNIX) and nmake (Windows) building procedures has been stopped.

Form of Open CASCADE Technology Delivery

- Starting from version 6.3. Open CASCADE Technology can be installed with binaries precompiled by Visual C++ 7.1 using Installation Procedure under Windows platform only. Automatic installation procedures and pre-compiled binaries for SunOS and Linux are not provided any more.
- Starting from Open CASCADE 6.3 the source package of the Open CASCADE Technology (including the source files of samples, 3rd party tools, and a set of building procedures) is available for UNIX and Linux (and Windows, as well) platforms.

  The building tools are delivered in the form of Visual C++ Projects for Windows platform and a Makefile procedure for Unix platform.



Directory Tree

```
OpenCASCADE
  data
  doc
  ros
    adm
    drv
    inc
    src
  samples
    standard
      c#
      java
      mfc
      qt
    tutorial
```

Description of the Directory Tree

data - This folder contains files different formats for experiments with Open CASCADE functionality

doc - This folder contains Open CASCADE Overview documentation;

ros/adm/make - This folder contains files of Makefile procedure , which allow rebuilding Open CASCADE on Unix platforms;

ros/adm/win32 - This folder contains Visual Studio workspace for Visual C++ 6.0, 7.1 and 8.0, which allows rebuilding Open CASCADE under Windows platform in 32 bit mode.

ros/adm/win64 - This folder contains Visual Studio workspace for Visual C++ 8.0, which allows rebuilding Open CASCADE under Windows platform in 64 bit mode.

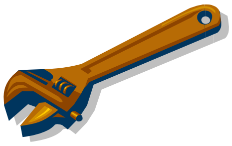
ros/drv - This folder contains source files generated by WOK (private header files and instantiations of generic classes);

ros/inc - This folder contains all Open CASCADE header files;

ros/src - This folder contains Open CASCADE source files. They are organized in folders, one per development unit;

Samples - This folder contains source files and building procedure for Standard samples of applications and Tutorial (c# and mfc samples are applicable only for Windows platform);

Bug Fixes



- Since last minor release (version 6.2) Open CASCADE 6.3 incorporates **170** modifications (bug fixes, enhancements and other corrections). For details, refer to **Appendix**.

Appendix: Open CASCADE 6.3 Modifications

- Foundation Classes
- Modeling Data
- Modeling Algorithms
- Visualization
- Application Framework
- Data Exchange
- Shape Healing
- WOK
- Draw
- Development Environment

Products

- OMF
- Express Mesh
- Parasolid interface
- ACIS interface
- Collision Detection

	Foundation Classes, 32 modifications
ID	**Short Description**
14670	Method to access object contained in the map
15125	Ncolleciton_Arra2 returns wrong values by RowLength and ColLength methods
15159	Bug in the method Subtract of the class TcolStd_PackedMapOfInteger
15190	Locking in Standard_MmgrOpt
15489	Constructor gp_Lin2d(A, B, C) creates line with origin point in infinity
15679	Standard_Type::Size() method returns the string length of the class name instead of the sizeof(class)
15899	Problem with launching DRAWEXE on IRIX 32-bit
16484	New data structures for memory- and performance-critical applications
16485	Bnd_Box method Add(Box) incorrectly takes Gap into account
16494	Reimplement Message mechanism according to last modifications in projects
16602	[OCC Forum] Integration of code porting to MacOSX by Torsten Sadowski
16618	Minor corrections in several classes (AIS, TcolStd, PrsMgr, OSD)
16689	Suppress redundant debug message in NCollection_UBTreeFiller
16832	New methods in Bnd classes (Bnd_B2x and Bnd_B3x)
16846	[OCCT Forum] Multithreading troubles: reference counter of Standard_Transient
17133	Correction of OSD_signal.cxx
17654	Minor corrections in several classes (Standard, Message, TcolStd)
17861	WNT MS VC8.0: conversion of big double value to int raises signal not handled by OSD::SetSignal()
18364	Improve `NCollection_CellFilter` to have different cell size on each dimension and to allow removal of objects from the structure
18408	Reading invalid BRep file
18540	Memory leak in OCC

18984	Bug in the method Subtract of the class TColStd_PackedMapOfInteger (#2)
19186	Eliminating of compilation problem.
19279	Corrections in OCCT headers to avoid compiler warnings
19308	Make consistent Standard_CString and Standard_ExtString definition.
19640	Change the visibility for `math_IntegerVector` copy-constructor
19984	Data structure supporting mesh analysis and modifications
19987	Using atan2 function in calculation of angles between 2 vectors
20005	Performance improvements of NCollection containers
20017	Eliminating of compiler warnings
20178	Conversion problems during VS2005 compilation
20210	Compatibility problems in OCC6.3 pre-release
	Modeling Data, 2 modifications
ID	**Short Description**
9303	Intersection curve surface doesn't take account of bounds of the surface
13028	IntAna2d_AnaIntersection(const gp_Lin2d&,const gp_Circ2d&) ignores circle sense
	Modeling Algorithms, 46 modifications
ID	**Short Description**
11042	Exception in ChFi3d_Builder, function ChFi3d_PerformElSpine
11565	Boolean operation "Cut" in 2d cannot cut the hole from the rectangle
12507	Wrong result of fuse operation
13538	Problem with Boolean operation on Shells
13904	Exception during "filling" operation
14376	Shading triangulation of face is not computed
14643	Checkshape command gives wrong result for compound of 66 solids
15500	Bad work nbshapes draw-command
15515	Integration of improvement of volume computation in Open Cascade
15519	Exception while meshing shape
15804	Restructuring of Boolean Operations algorithm. Part# 1
15836	Wrong visualization of filleted shape on IRIX32 platform
15850	Regression in BOP – wrong results for simple shapes with Bspline geometry
15936	Wrong shape build by revolution algorithm
15943	Wrong result of boolen fuse.
15968	Result of checkshape command depends on order of subshapes in a shape
16119	Bug in GeomFill_Coons algorithm
16179	Printing of error dump by Fillet algorithm
16517	Cylindrical projection is wrong
16662	Crash in ShapeAnalysis_Wire::CheckSmall
16667	2D Offset algorithm fails
16781	Wrong result of Cut operation.
16833	Error in Coons algorithm
16847	Integration in CASCADE FIP "Topological refiner"
16852	Error in Extrema_ExtPElC2d::Perform
17046	Exception in Extrema_ExtPS on Mandriva2006 32-bits
17357	Any Boolean operation is impossible between attached shapes
18082	The method Standard_Boolean Bnd_B2x::IsOut(const gp_Ax2d& theLine) const works

	incorrectly
18541	Problem with GCPnts_UniformAbscissa class
18912	Unification of BRepMesh_Discret and BRepMesh_FastDiscret algorithms
18961	Min distance is defined incorrectly between 2 objects
19071	The MakeFuse or the MakePartition function crash together on 2 torus.
19559	Progress indicator in BRep reader / writer
19667	Appendix for Salome bug NPAL19578.
19751	Integration of BRepTools:Update performance improvement
19793	Fuse problem of symmetrical shapes.
19811	Scaling problem.
19887	License problem on 64 bit platforms
20029	Bounding box gives strange values on a sphere
20037	Shading display on a spring is very bad
20055	Command distmini produce wrong result for NT and SUN in optimize mode
20056	The bug is appendix to the Salome Bug 0019881.
20061	Error on Create shape with preview when Shading mode is on.
20093	Development of Surfaces
20104	The bug is appendix to the Salome Bug 0019912.
20196	Problem with DRAW commands that use pick as arguments.

	Visualization, 37 modifications
ID	**Short Description**
7721	Visualization of 33 000 interactive objects crashes an application
12584	Exception "Invalid Floating Point Operation" in `V3d_View::ColorScaleDisplay()` method
13439	The text in ColorScale is not displayed on Linux.
14420	Bnd_BoundSortBox2d works incorrectly if all objects are degenerated into a line parallel to X axis of viewport
14821	Regression in anti aliasing
15196	MeshVS: Implement selection mechanism with single owner for the whole mesh
15213	Incorrect conversion of 2d coordinates of reference plane into pixel coordinates in side view
15571	Improve MeshVS_VectorPrsBuilder to show simplified arrows
15671	Mistakes in CDL files
15793	Provide access to OpenGl depth buffer from applications using OCC viewer
15988	Crash in AIS_DiameterDimension for non-automatic positioning
15989	Symbol of the diameter partially hides the diameter value in the Viewer 3D
16050	Transparency for Visual3d_Layer
16055	Support of groups in MeshVS
16207	Problem of displaying AIS interactive object on Unix
16300	MeshVS: custom presentation is lost if it should be drawn by PrsBuilder of second priority
16487	Aspect_TypeMapEntry initialization
16786	Contribution for UserDraw mode amelioration (ZbufferAuto feature)
16798	OpenGL: text optimization
16950	OCC Visualization fails to display empty nested compounds
16981	Fonts are broken if the application closes all windows before opening a new window.
17656	Loading attached BRep file in 3D Viewer raises hanging of the latter
17899	OCC Forum: Select3D_Box2d does not handle "void" correctly
18349	MeshVS mesh presentation builder optimisation
19148	A narrow pipe is badly shaded

19373	Add API for AIS_InteractiveContext to get Erased objects
19410	MeshVS_Tool has uninitialized variables
19433	SetAxialScale method of V3d_View rise an exception if view is empty.
19639	Integration of Layer Manager
19820	3D discrete topology (voxels)
19821	3D discrete topology (voxels)
19862	The Draw axonometric view in draw can not be saved in gif format.
19915	Problems with transparency
20035	Addition of AVI recorder to TKOpenGl
20053	Introduction of NIS vizualization library
7691	Wrong hidden lines computed by HLRBRep_PolyAlgo and HLRBRep_PolyHLRToShape in OCC 5.1
18942	Improved perspective in OCCT 3D view
	Application Framework, 21 modifications
ID	**Short Description**
1138	OSD_SIGSEGV is thrown during call LDOMParser.parse method
5032	LDOM parser and XML formatter should be able to treat UNICODE strings
8988	Document open/save functions do not work with simple file name
9745	In BinTools_ShapeSet, methods WriteGeometry and ReadGeometry are inconsistent
9746	Incorrect writing of integer array in BinMNaming_NamingDriver::Paste
10138	Crash on attempt to copy an OCAF attribute: TDataStd_RealArray, TDataStd_IntegerArray
16495	Create reusable intermediate layer between OCAF and GUI
16497	XML OCAF storage of document on disk looses "0" in "00", "000", etc.
16745	Resume of an attribute raises an exception
16748	Reference to an empty label is lost on open of a document
16782	New OCAF attributes
16879	Storage of a document on disk into a read-only file fails, but no error is issued
16965	OCAF attributes should return strings by reference rather than by value
17140	Copying a presentation to another presentation is not very clear
18570	A regression in TDataStd_RealArray::Set()
19184	Bad performance of Append operation of TDataStd_TreeNode class
19403	Adding DeltaOnModification functionality to set of Standard attributes
19592	Advancement of Function Mechanism of OCAF
19847	NamedShape presentation driver (TPrsStd_NamedShapeDriver) displays not expected shape
19986	Version "3" of binary OCAF persistence (BinOcaf)
20008	Binary format is incompatible between 32 and 64 bits OSs
	Data Exchange, 15 modifications
ID	**Short Description**
13542	Export to BRep, IGES, STEP with not enough space on disk: file is invalid, but no error status returned.
15220	Problems with IGES file locations
15570	Incorrect handling of Unicode strings
15755	IGESData_IGESEntity::NameValue returns nothing when nbname > 1
16351	Crash in ShapeFix_Edge::FixAddPCurve
16569	Exception in IGESDraw_Planar::Init when allEntities parameter is a null handle

17026	Problem of reading IGES files
17099	[OCC Forum] bug in reading IGES file
17927	NaturalRestriction flag fails after sewing
18612	Exception is raised during reading big file (file xyz.igs)
19110	Reading IGES file hangs
19586	Reading "NOT VISIBLE" entities from IGES
19777	Integration of new version of read/write VRML2.0 files
19990	Exception on reading of validation properties
20054	Appearance node not available for VRML primitives (box, cylinder, etc.)
	Shape Healing, 1 modification
ID	**Short Description**
19764	'Out of Memory' exception during reading of an IGES file
	WOK, 1 modification
ID	**Short Description**
19770	Implementation of returning pointer to OCC Handle for performance reasons.
	DRAW, 2 modifications
ID	**Short Description**
19946	Draw command "whatis ." does not print the name of geometry object picked on a Draw viewer
20196	Problem with DRAW commands that use pick as arguments.
	Development Environment, 4 modifications
ID	**Short Description**
15767	Portability CASCADE 6.2 to gcc4.1.X
18669	make install of OCCT installation works incorrect
18670	64 platforms should be detected automatically by configure script of OCCT
20184	Problem of make install 2 times

Product Bug Fixes

The following bug fixes have been performed for Open CASCADE products customers.

	OMF, 3 modification1
ID	**Short Description**
16648	Integration of improvements of NASTRAN reader and NASTRAN writer.
18081	Invalid result of Boolean operations
20211	Corrections in NASTRAN Formatter class for OCC 6.3 release

	Express Mesh, 1 modification
ID	**Short Description**
19587	ExpressMesh ignores some holes in a shape
	Parasolid interface Interface, 1 modification
ID	**Short Description**
15225	Incorrect usage of TCollection_AsciiString::Insert method
	ACIS interface Interface, 3 modifications
ID	**Short Description**
16327	Incorrect handling of tcoedge, tedge, tvertex ...
16483	Reading of SAT Versions 10.0 - 14.0
20098	Problems with # in string constants and improvement of fix natural bounds
	Collision Detection, 1 modification
ID	**Short Description**
16215	Additional checks; thread-safe intersector