Open CASCADE 6.2.1 Maintenance Release

Release Notes

Overview

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

Version **6.2.1** 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 6.2.1.

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**
  - **Performance and functional improvements in MeshVS package**
  - **New advanced selection mode in MeshVS for efficient work with big meshes**
  - **New standard attributes included in OCAF**
  - **Integrated code changes made by OCC users for MacOS X and FreeBSD porting**
- **Products**
  - **Reading of ACIS SAT versions 10.0 - 14.0**
  - **Improved tools for reading and writing NASTRAN files in OMF**
  - **New tools for export of 2D geometrical data to DXF**

New Features

Foundation Classes

- 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

Modeling Algorithms

- New functionality provides shape volume computation by adaptive Gauss-Kronrod method. New computation methods are accessible via new methods of class `BRepGProp: VolumePropertiesGK(...)`.

Visualization

- 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 100K.

  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.
- The possibility to set the transparency for the objects of `Visual3d_Layer` has been added.

Application Framework

- 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`.

Draw

- 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

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).

DXF

- A general mechanism for writing 2D DXF entities has been implemented. The basic DXF packages (DXFControl) have been extended.

Improvements

Foundation Classes

- 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_Lin(A, B, C)` 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.

Modeling Algorithms

- 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.

Visualization

- 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 OCCT5.2.3 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.

Application Framework

- 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".
- 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.

Data Exchange

- The methods for writing shape in the file for STEP, IGES and BREP formats have been improved to check the state of ofstream 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.

Products

OMF

- NASTRAN reader has been improved and is now able to handle properly different types of continuation entries.

Parasolid

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

ACIS SAT

- Processing of ACIS SAT topological entities possessing tolerance (`tedge`, `tvertex` and `tcoedge`) has been improved. Earlier they were not taken into account.

Changes

Foundation Classes

- 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 <<")

Visualization

- 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 presented in Open CASCADE)

Data Exchange

- 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.

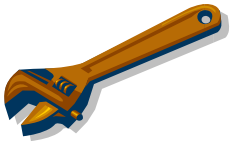
Building tools

- MS Visual Studio project 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 Open CASCADE using either VC++6.0, VC++7.1 or VC++8.0.
- New flags have been added to Open CASCADE Makefile procedure. Now you can build Open CASCADE without Draw, WOK and Jcas.

  Additional flags:
  - --disable-draw - allows Open CASCADE building without Draw.
  - --disable-wok - allows Open CASCADE building without WOK.
  - --disable-jcas - allows Open CASCADE 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-jcas=yes --with-java-include=`/path-where-java-included-are-installed

Bug Fixes



- Since last minor release (version 6.2) Open CASCADE 6.2.1 incorporates **87** modifications (bug fixes, enhancements and other corrections). For details, refer to **Appendix**.

Appendix: Open CASCADE 6.2.1 Modifications

Products

	Foundation Classes, 16 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
	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, 25 modifications
ID	**Short Description**
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
16852	Error in Extrema_ExtPElC2d::Perform
17046	Exception in Extrema_ExtPS on Mandriva2006 32-bits
17357	Any boolean operation is impossible between attached shapes
	Visualization, 18 modifications
ID	**Short Description**
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.
	Application Framework, 12 modifications
ID	**Short Description**
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
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
	Data Exchange, 8 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
	Development Environment, 1 modification
ID	**Short Description**
15767	Portability CASCADE 6.2 to gcc4.1.X

Product Bug Fixes

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

	OMF, 1 modification
ID	**Short Description**
16648	Integration of improvements of NASTRAN reader and NASTRAN writer.

	Parasolid, 1 modification
ID	**Short Description**
15225	Wild regression on OCC 6.2
	ACIS, 2 modifications
ID	**Short Description**
16327	Incorrect handling of tcoedge, tedge, tvertex ...
16483	Reading of SAT Versions 10.0 - 14.0
	DXF, 1 modification
ID	**Short Description**
16388	Open CASCADE improving for DXF2D connector

