

Open CASCADE 5.2

Minor release



Release Notes

Overview

Open CASCADE Technology 5.2 is a minor release, which includes new features, improvements and bug fixes, over minor release 5.1 and maintenance releases 5.1.1, 5.1.2 and 5.1.3*.

Version 5.2 is binary incompatible with any of the previous versions of Open CASCADE Technology, so applications linked against a previous version (major, minor or maintenance) must be recompiled to run with this Version 5.2.

Table of Contents

- **Highlights**
- **New Features**
 - [Application Framework](#)
 - [Data Exchange](#)
 - [Visualization](#)
 - [Programming Samples](#)
 - [Installation](#)
 - [Test Harness Draw](#)
- **Improvements**
 - [Technical Documentation](#)
 - [Application Framework](#)
 - [Data Exchange](#)
 - [Foundation Classes](#)
 - [Modeling Algorithms](#)
 - [Modeling Data](#)
 - [Visualization](#)
 - [WOK](#)
- **Changes**
 - [Support of STL Streams](#)
 - [Standard MFC Samples](#)
 - [Data Exchange](#)
 - [Application Framework](#)
 - [Modeling Algorithms](#)
 - [Visualization](#)
 - [Installation](#)
 - [Building Tools](#)
 - [Test Harness Draw](#)
 - [Qt Samples](#)
 - [Porting to VC7](#)
- **Bug Fixes**
- **Appendix 1 (Bug Fixes)**
- **Appendix 2 (Removed Entities)**
- **Appendix 3 (Test Harness Draw Changes)**
- **Appendix 4 (New TH Draw commands)**

* Available for A la carte, E-learning and Ten Query Support package customers. See our news at <http://www.opencascade.org/about/news/issue112/> and <http://www.opencascade.org/about/news/issue113/> for details.



Highlights

- **OPEN CASCADE NOW SUPPORTS MS WINDOWS® XP, RED HAT LINUX 8.0 AND MANDRAKE LINUX 8.0 !**
- **DRAW TEST HARNESS HAS BEEN MADE A SINGLE EXECUTABLE – DRAWEXE WITH DYNAMICALLY LOADED COMMANDS;**
- **TWO NEW OCAF PERSISTENCE SCHEMAS – XML AND BINARY – HAVE BEEN ADDED;**
- **NEW COMPONENT FOR VISUALIZING MESHES- MESHVS;**
- **STL STREAMS SUPPORT HAS BEEN ADDED;**
- **SUPPORT OF MICROSOFT .NET HAS BEEN ADDED AND DEMONSTRATED WITH THE HELP OF THE NEW C# PROGRAMMING SAMPLE;**
- **DOCUMENTATION ON PIPES, OFFSETS HAS BEEN EXTENDED. JAVA EXTRACTOR DOCUMENTATION HAS BEEN ADDED;**
- **TWO MORE OPEN CASCADE TECHNOLOGY ALGORITHMS (LOCAL OPERATION AND 3D OFFSETS) HAVE BEEN PORTED TO NEW BOOLEAN OPERATIONS THEREBY GAINING BETTER QUALITY AND RELIABILITY;**
- **TEXT OUTPUT CAPABILITIES OF OPEN CASCADE 3D VIEWER (WINDOWS PLATFORM ONLY) HAVE BEEN EXTENDED WITH THE POSSIBILITY TO USE TEXTURE-MAPPED FONTS AS AN ALTERNATIVE TO TRADITIONAL BITMAP FONTS;**
- **AN IMPORTANT IMPROVEMENT IN VISUALIZATION: NOW A LARGE NUMBER OF OBJECTS (UP TO THOUSANDS) CAN BE HIGHLIGHTED IN THE 3D VIEWER;**
- **XDE (EXTENDED DATA EXCHANGE) READER HAS BEN EXTENDED TO READ INFORMATION ABOUT MATERIALS AND GEOMETRIC DIMENSIONS AND TOLERANCES FROM STEP FILES;**
- **VARIOUS PERFORMANCE IMPROVEMENTS ESPECIALLY IN STEP AND IGES DATA EXCHANGE AND OCAF.**

New Features

Application Framework

- Redirection of all messages from Persistence has been implemented. This allows to use the so-called application Message Driver (if defined) for output of all warnings and any other relevant messages. If the Message Driver is defined, all warnings, additional messages will be directed to this Driver. If the Message Driver is not defined, the default CDM_NullMessageDriver will be used to process all messages. In that case no messages are shown or passed to higher-level routines.

Example:

This is an example of the driver activation:

```
CDM_MessageDriver aDriver = new CDM_CoutMessageDriver();
myApp->SetMessageDriver (aDriver);
```

The following example (taken from PCDM_ReadWriter_1.cxx) shows how to use the already set driver to output custom messages:

```
TCollection_ExtendedString aMsg("warning: ");
aMsg.Cat("could not read the reference counter in
").Cat(aFileName).Cat("\0");
if(!theMsgDriver.IsNull())
theMsgDriver->write(aMsg.ToExtString());
```

- A new ExtStringArray class has been added to the TDataStd package. This class is intended to provide the implementation of a new standard attribute to keep the array of Unicode strings within the Data Framework at a predefined label. For more advanced user information, please, consult OpenCASCADE Reference Documentation (chapter Standard Attributes).
- In order to work-around incorrect work of the putenv function of Tcl on Windows, the following files have been modified:

Resources_Manager.cd1; Resource_Manager.cxx

A new constructor of Resource_Manager has been added:

```
Resource_Manager(const Standard_CString aName, TCollection_AsciiString&
aDefaultsDirectory, TCollection_AsciiString& anUserDefaultsDirectory, const
Standard_Boolean Verbose = Standard_False);
```

where:

aDefaultsDirectory and anUserDefaultsDirectory are the directories for resource files aDefaultsDirectory/aName and anUserDefaultsDirectory/aName.

- **XML and Binary persistence**

XML and Binary persistence schemas have been implemented in addition to Standard persistence schema. All schemes are independent of each other, but they guarantee that the standard OCAF attributes stored and retrieved by one schema will be convertible to another. Therefore in any OCAF application you can use any persistence schema or even all.

- Standard persistence. Its identifiers are MDTV-Standard (for basic attributes) and MDTV-XCAF (for XDE attributes). It stores all information in ASCII using a special text format.
- XML persistence. Its identifiers are `Xm\Ocaf` and `Xm\XCAF`. It stores information using the XML format that can be read by means of any XML editor or viewer.
- Binary persistence. Its identifier is `BinOcaf` (implementation of XDE attributes will be made available in further releases). It stores information in the binary format that provides better performance. Binary files can be read by OCAF on another platform due to a portable schema of read/write operations.

Data exchange

- Timers have been implemented that are used for measuring the performance of a current operation or any part of code, and provide the necessary API. Timers are used for debugging and performance optimizing purposes. See Open CASCADE Reference Documentation (MoniTool Package) and Shape Healing User's Guide for more details.
- A new method: `ShapeFix_Face::FixIntersectingwires()` has been created. It detects and fixes the specific case when a face has more than one wire and these wires have an intersection point. In this case a common vertex is assigned (either new or an existing one) for the intersection of wires in the point of intersection.
- In the framework of regular CAX-IF* activity, Open CASCADE has extended its STEP Interface to handle STEP materials, dimensions and geometric tolerances.
- The **Progress Indicator** functionality, including a possibility of operation interruption by the user (break), has been added to IGES and STEP translators and Shape Healing.

The new class `MoniTool_ProgressIndicator` is provided for applications to facilitate indication of progress of time-consuming procedures and providing a possibility for the user to break the procedure. It is an interface class implementing a virtual progress bar that can be incremented in several ways. In particular, it supports hierarchical incrementing in case of arbitrarily nested procedures, and provides optional logarithmic progress scale. The progress indicator is designed to be easy to use and induce minimal impact on application performance. See more details in the CDL file of the `MoniTool_ProgressIndicator` class.

To show a progress indicator in an application, it is necessary to derive the application-specific class implementing a method for visual feedback of progress according to the application, and (optionally) a method to provide the user break command. Then this class should be passed to progress indication-enabled procedure.

As an example of usage of progress indicator interface, please refer to IGES and STEP translators (see, for instance, class `STEPControl_ActorRead`). The example implementation of the progress indicator in DRAW can be found in class `XSDRAW_ProgressIndicator`. To see how it works (it is

* The CAX Implementer Forum (CAX-IF) is a group of software developers and testers working on CAD translators based on the ISO 10303 STEP. See also http://www.cax-if.org/who_we_are.html



disabled by default), run command **XProgress +g** in DRAW before running commands to read or write STEP or IGES files (e.g. **igesread**).

Visualization

- In order to increase performance when working with large selections the previous approach of highlighting selected entity owners has been extended to support a smarter behavior. New methods for highlighting selected and detected entity owners in the 3D viewer have been introduced:

- AIS_LocalContext::UpdateSelected(const Handle(AIS_InteractiveObject)&, Standard_Boolean)
- SelectMgr_EntityOwner::IsAutoHighlight()

SelectMgr_SelectableObject class.

Methods:

- HighlightSelected(),
- ClearSelected(),
- HighlightOwnerWithColor(),
- IsAutoHighlight(),
- SetAutoHighlight() have been added.

An optional boolean argument has been added: ClearSelections(const Standard_Boolean update = Standard_False)

Please, see Reference Documentation for full details.

- Texture mapped fonts are now available on Windows platform as an alternative and more efficient method for text output in Open CASCADE 3D viewer.

[See below for more information](#)

- **MeshVS** (Mesh Visualization Service) component has been added that extends 3D visualization capabilities of Open CASCADE Technology. It provides flexible means of displaying meshes along with associated pre- and post-processor data.

From a developer's point of view, it is easy to integrate the MeshVS component into any mesh-related application with the help of the following guidelines:

1. Derive a data source class from the MeshVS_DataSource class. Re-implement its virtual methods, so as to give the MeshVS component access to the application data model. This is the most important part of the job, since visualization performance is affected by performance of data retrieval methods of your data source class.
2. Create an instance of the MeshVS_Mesh class.
3. Create an instance of your data source class and pass it to MeshVS_Mesh object through the SetDataSource() method.
4. Create one or several objects of MeshVS_PrsBuilder-derived classes (either standard, included in the MeshVS package, or your custom ones). Each PrsBuilder is responsible for drawing a MeshVS_Mesh presentation in certain display mode(s) specified as a PrsBuilder constructor's argument. Display mode is treated by MeshVS classes as a

combination of bit flags (two least significant bits are used to encode standard display modes: wireframe, shading and shrink). Pass these objects to the `MeshVS_Mesh::AddBuilder()` method. `MeshVS_Mesh` takes advantage of improved selection highlighting mechanism: it highlights its selected entities itself, with the help of so called "highlighter" object. You can set one of `PrsBuilder` objects to act as a highlighter with the help of a corresponding argument of the `AddBuilder()` method.

5. Visual attributes of the `MeshVS_Mesh` object (such as shading color, shrink coefficient and so on) are controlled through `MeshVS_Drawer` object. It maintains a map "Attribute ID --> attribute value" and can be easily extended with any number of custom attributes.
6. In all other respects, `MeshVS_Mesh` is very similar to any other `AIS_InteractiveObject`-derived class, and it should be used accordingly (refer to the description of AIS package in the documentation).

For Open CASCADE Technology particular implementation of MeshVS to read/write STL files see the `ViewerTest` package and refer to [Appendix 4](#) describing new commands included into Test Harness.

Programming Samples

- New Import / Export sample. Sample demonstrating the use of .NET technology with C# language in Open CASCADE Technology.
- Extension of `graphic3ddemo` sample. Demonstration of advantage of using texture-mapped fonts as an alternative to traditional bitmap fonts.
- Extension of `Viewer 3D` sample. Demonstration of improved displaying of overlapping or touching surfaces in the shading mode, color gradient filling and trihedron.

Installation

- The installation procedure is now cross-platform. This makes more comfortable the procedure of installation of Open CASCADE Technology package on all supported platforms - Windows, Sun or Linux.

Test Harness Draw

- Open CASCADE Test Harness has been improved. The improvement allows dynamic loading of a corresponding group of Draw commands using only one version of Draw executable - **DRAWEXE**. All numerous instances of Test Harnesses for Open CASCADE modules (such as TCAF for OCAF, XSDRAWEXE and XDEDRAWEXE for Data Exchange, etc) have been eliminated. This executable is now capable of handling arbitrary commands loaded as plug-ins at run-time.

For details, please refer to the User's Guide and [Appendix 3](#).

- Test 3D view implemented in the `ViewerTest` package has been improved. Now it provides such common user interface feature as selection with a rectangle.



Improvements

✓ Technical documentation

- WOK User's Guide has been extended to describe the use of Java extractor. Refer to the User's Guide for full details.
- Reference Documentation has been updated, including the description of Pipes and Offsets.

✓ Application Framework

- An improvement has been made which is internal to package TDF and class TDF_AttributeIterator. The type of the myValue field of the TDF_AttributeIterator class has been changed from a smart pointer Handle(TDF_Attribute) to a standard pointer TDF_Attribute*. This improvement allows to gain about 25% performance increase especially when working with extensive OCAF documents.

1. Method TDF_AttributeIterator::Value now returns (TDF_Attribute *), previously it returned Handle(TDF_Attribute). This is done to avoid creation/copying of Handle by this method. Generally this change does not require modification of the existing application code (pointer is implicitly casted to Handle).
2. The AttributeIterator object no longer holds the current OCAF attribute as a possessor of a reference to it (via the mechanism of Handle). This means that the attribute may be destroyed when the Iterator is still active, and the call of the method Next() would cause unpredictable results.

In practice, this can only happen when the method TDF_Label::ForgetAttribute is called inside the loop managed by a corresponding AttributeIterator instance. To avoid such problem you should always call Next() BEFORE any possible calling of TDF_Label::ForgetAttribute, like in method TDF_Label::ForgetAllAttributes:

```
TDF_AttributeIterator itr1 (myLabelNode);
// OCC5031: iterator must be incremented before removal of the attribute
while (itr1.More()) {
    const Handle(TDF_Attribute) anAttr = itr1.Value();
    itr1.Next();
    ForgetFromNode (myLabelNode, anAttr);
}
// Previous code (dangerous):
// while (itr1.More()) {
//     ForgetFromNode(myLabelNode,itr1.value());
//     itr1.Next();
// }
```

- **TDataStd_IntegerArray and TDataStd_RealArray TDataStd_ExtStrinArrayArray classes**

Extra backups for an attribute have been removed if Value is not really changed. Backup is performed only if:

- dimension of <NewArray> differs from that of the old Array (<myValue>).
- at least one element of <newArray> differs from a correspondent element of <myValue>.

If the Backup is performed, a new instance of HArray1of...(HArray1ofInteger or HArray1ofReal) is created and this instance consequently handles <myValue> if current and backed up attributes are different.

- **TDocStd_MultiTransactionManager class**

A possibility has been added to detect whether a just performed transaction changes something in a document and writes this changed information into the undo buffer.

- **LDOM package and XML persistence**

Now a string containing non-ASCII characters is correctly retrieved from an XML file. The operator of converting an object of type LDOMBasicString to TCollection_ExtendedString takes care of such characters.

- The memory occupied by a temporary relocation table is freed at the end of the store/retrieve operation.

- The robustness of the **LDOM package** has been increased:

- Some memory leak has been eliminated;
- Reading of memory beyond the allocated space has been prevented, as well as reading of non-initialized memory.

- The Naming mechanism has been extended to cover cases when several entities are considered as generators.

- A possibility has been added to remove drivers from the static maps TPrsStd_DriverTable and TFunction_DriverTable with the help of methods:

- RemoveDriver - removes a driver with the given GUID,
- Clear - removes all drivers.

This improvement allows to :

- Avoid problems connected to the unloading order of modules where the code of custom drivers is defined. Previously an exception could be raised if a custom module had been unloaded before the destruction of tables.
- Change presentation and function drivers at any point in the program thus implementing some sophisticated behavior of an application.

Note:

If the TPrsStd_DriverTable::Clear method is used, the InitStandardDrivers should be called to fill the table with standard drivers.

- **Method TDocStd_XLinkTool::Copy()**

If the source label contains the TDataStd_TreeNode attribute then this method remembers its parent, first child, brothers and removes these entities from the tree of TreeNode before copying. After copying restores this attribute in the tree. Same for target label.

- Some unused classes and methods have been removed mainly from the CDF package. See **Appendix 2** for details.



Data Exchange

- Translation of some STEP and IGES files has been sped up. The resulting average acceleration for STEP is about 22% and for IGES it is about 19% over Open CASCADE version 5.0.
- The reading of B-Splines weights during an IGES translation has been improved. Some systems write the values of weights incorrectly (zero or negative values). This is because these systems dump "memory garbage" in case of polynomial B-Splines. Now all such weights for this type of BSplines are set to 1.
- A fix has been added for faces having wires composed of several (intersecting or touching) loops of edges. For such cases the fix result is a face with a set of internal wires obtained from the initial wire, or this fix result is a set of faces whose outer wires are obtained from each edge loop.
- A face orientation fix has been added into ShapeConvert_UnionEdges to be sure that a face does not become invalid.
- A new fix for correction of invalid positions of vertices has been added. This fix is activated by setting the flag `FixShape.FixVertexPositionMode` to -1 or 1 (zero by default) in the resource file. This fix checks the tolerance of vertices and if this tolerance exceeds the tolerance specified by the user, the vertex moves to the barycenter of the points corresponding to the ends of edges referring to this vertex.
- A face orientation fix has been added to move parametric curves of internal wires into face boundaries.

Foundation Classes

- A memory leak has been eliminated in the concatenation function (`Cat`) of the string types represented by classes `AsciiString` and `ExtendedString` from the `TCollection` package.
- **NCollection package**
The behavior of the `Bind` method of the `DataMap` class has been corrected so that it corresponds to its counterpart in the `TCollection` package. In particular, the method must bind the new value to a given key even if the same key is found in the map.
- Some improvements have been made in the `NCollection` package to make it more flexible:
 - Methods `NCollection_List::Append()` and `Prepend()` have been modified to return the reference to a newly added item.
 - A new method `NCollection_IncAllocator::Reallocate()` has been added. It allows to resize the most recently allocated memory block.
 - Method `NCollection_BaseMap::Iterator::Initialize()` has been made public. It allows a developer to reuse an already used iterator object once again with a different map object.
 - The compilation warning C4291 ("no matching operator delete found...") of MS Visual Studio has been eliminated.

- **Standard Memory Manager**

It has been optimized to work with memory-checking tools such as Rational Purify. Now the deactivation of the standard optimized mode (through environment variable MMGT_OPT) makes the memory manager work directly through malloc and free and do not reserve extra space for the memory block header.

- **OSD package**

A problem has been fixed when due to a different system API on Win9x, method SystemTimeToTzSpecificLocalTime did not perform any time conversion, it just left all fields of the output parameter structure non-initialized, leading to application crash.

- The specific (stack-like) memory allocation mechanism used in array classes from the math package (such as math_RealVector) is replaced by usage of standard Open CASCADE memory manager (from package Standard). This replacement results in memory usage optimization and allows to avoid potential memory leaks (that could previously happen in case of allocation of big amounts of memory in several arrays simultaneously).

Modeling Algorithms

- The Boolean operations algorithm has been corrected to process the computation of a 2d curve corresponding to a 3d curve at the intersection of faces. No exception is raised now during the cut operation.
- Some limitations of New Boolean operations concerning the impossibility to process shapes with large tolerances of edges and vertices are solved, including cases where the processed shapes have same domain faces and same domain edges. Additional topological verification to perform (if possible) an immediate correction of building result has been implemented into the New Boolean operations algorithm.
- Projection of a 3d point on a surface of revolution is now computed correctly for the case when the axis of revolution does not lie in the plane of the revolved periodic curve.
- The shape analyzer now processes tiny faces more correctly than before.
- The problem in the fillet algorithm caused by the variation of input data precision influencing the result of calculation has been fixed.
- The BRepOffsetAPI_MakePipeShell algorithm can now process pipes in the sweep mode set either to IsGuideACWithContact or to IsGuidePlanWithContact.
- Now the algorithm of BRepOffsetAPI_MakePipeShell correctly processes the cases with linear parts of spines and profiles.
- The BRepOffsetAPI_MakePipe algorithm has been improved. The parameter of a given path and corresponding to a section, is now defined correctly, so the resulting swept surface is now built correctly too.
- **The Surface/Surface Intersection algorithm has been improved**
 - The stability and quality of the Surface/Surface Intersection algorithm (SSA) has been improved, as well as results of all basic modeling algorithms of Open CASCADE Technology that use the SSA (Boolean Operations, Partition, Gluer, Fillets, Features, etc.)

- The SSA has been supplied by a post processing part that allows to split curves going through the apex and seams (natural boundary of a periodical surface). The post processing algorithm corrects bounds of curves to make them smoother.
- The post-processing part is generalized. It can be used (with small modifications) as one common code for the SSA and new Boolean Operations Algorithm where this part is a single part.
- A new algorithm for finding interference between surfaces has been designed and developed. The algorithm allows to find interference between surfaces for cases where the old algorithm fails, and decreasing computation time for some cases (tenfold and more!) of intersection.

- Now the SSA algorithm can process the case of two previously non-processable trimmed toruses.
- The problem of building result on edges with the same locations has been eliminated.
- Invalid behavior in memory management in new Boolean Operations has been fixed.
- The BRepExtrema_DistShapeShape algorithm for searching the shortest distance between shapes has been optimized.
 - An empty (default) constructor has been additionally created for the class BRepExtrema_DistShapeShape.
 - Two methods have been added to the BRepExtrema_DistShapeShape class that allow to change shapes in cycle when searching the shortest distances from one shape to a set of shapes.


```
LoadS1(const TopoDS_Shape& Shape1)
LoadS2(const TopoDS_Shape& Shape1)
```
 - Another method has been added to set the deflection (used to specify a maximum deviation of extreme distances from the minimum one) independently.


```
SetDeflection(const Standard_Real theDeflection)
```
 - Computation of the shortest distance is now processed by the function Perform.


```
Standard_Boolean Perform()
```
- Speed of **Boolean operations** has been increased for arguments based on parametric surfaces. This increase has been achieved due to increased performance of the intersection algorithm, which intersects a curve and parametric surface.
- Boolean operations have been improved for solid type of arguments. The improvement concerns processing of touching faces of the arguments. The improvements are in force if the section between arguments is computed correctly. Performance is also improved for the cases of computing different Boolean operations with the same arguments.

Modeling Data

- Class GCPnts_UniformAbscissa has been corrected in order to produce correct distribution of points on a composite curve with non-regular parameterization.



 **Visualization**

- **Visualization performance**

Now a developer can improve the performance of an application using a new optional boolean parameter of methods `Reset()` and `FitAll()` of the class `V3d_View` controlling whether an update is needed at the end of an operation.

- Performance of selection highlighting in 3D has been improved by introducing a standard sorting algorithm `SortTools_QuickSort` instead of a very rough algorithm used previously. The effect should be visible when there are many (up to thousands) entities under the mouse cursor.

- **AIS package**

Some methods now erase all object presentations for display modes different from the required one (passed as an argument to `SetDisplayMode()` or contained in the `<myDisplayMode>` field of an interactive object, in case of `Display()`) and then make sure that the object presentation for the given mode is displayed and may be highlighted (according to the current highlighting state of the object). Earlier it was impossible to switch to a required display mode in the field of an interactive object before calling `SetDisplayMode()`, because other presentations remained visible that led to a mess in the 3D viewer. So an application could not use the `<myDisplayMode>` field of an object to pass the mode information to some custom display mode manipulation logic. Now `SetDisplayMode()` does not rely on the display mode coming from the object field, so this field can be used safely by applications. `Display()` can now be applied to an object already displayed (formerly, it did nothing in such situation), to make sure that it is shown in a proper display mode (a convenient shortcut to combination of several method calls). It is necessary just to call `AIS_InteractiveObject::SetDisplayMode(mode)` and `SetToUpdate()` methods before calling `Display()` to make it work. It should be noted that calling `Display()` for an already displayed object activates the default selection mode for this object (usually 0), as it does for objects not yet displayed.

- Text output capabilities of Open CASCADE 3D viewer (Windows platform only) have been extended with the possibility to use texture mapped fonts as an alternative to traditional bitmap fonts.

This new feature makes text output on Windows several times faster as compared to the old approach, provided that graphic hardware and video drivers support hardware acceleration of texture operations. This is especially important for applications dealing with a large number of text labels displayed in the 3D viewer, since a 3D scene is updated very slowly with several thousand text labels displayed with the help of bitmap fonts.

For easy control over the new functionality, `CSF_TEX_FONT` environment variable can be used. Several values can be assigned to it having different meaning as described below:

less or equal to zero	texture mapped fonts cannot be used; only bitmap fonts are available;
1	texture mapped fonts are enabled, bitmap fonts are used by default (old text output method);
2	texture mapped fonts are enabled, texture mapped fonts are used by default (new method).

Hence to use the new text output method in an application, it is enough to set the `CSF_TEX_FONT` environment variable to 2, and no source code modification is needed. If the variable is not defined then use of texture mapped fonts is disabled and old approach is used.



To switch texture mapped fonts on/off in run-time, CSF_TEX_FONT should be set to 1 or 2, and Graphic3d_AspectText3d::SetTextureMappedFont (Standard_Boolean) method should be used with the Standard_True/Standard_False argument. First, a Graphic3d_AspectText3d object should be prepared using the above-described method, and then this object can be used in a usual manner, i.e. passed to the Graphic3d_Group::SetPrimitivesAspect() method. Without explicit calls to Graphic3d_AspectText3d::SetTexturedMappedFont() the objects get created with the value defined by the CSF_TEX_FONT variable (see the table above).

Note: This use convention (based on system variable) is temporary and will be changed in the next version of Open CASCADE Technology.

- Interface of AIS_InteractiveObject, AIS_InteractiveContext and Graphic3d_AspectFillArea3d classes has been extended to allow drawing of partially or fully overlapping polygons in the shading mode without any visual artifacts with the help of Open CASCADE Technology 3D graphic capabilities. Additionally, this improvement allows to avoid visual artifacts when drawing a curve lying in plane with some shaded polygon.

Usually, when you display two overlapping or touching surfaces or meshes in the shading mode, it results in "blinking" in coincident areas during 3D view rotation or zooming. This is default OpenGL behavior (the depth buffer simply cannot resolve which pixel of the two should be displayed).

Nevertheless, OpenGL provides a polygon offset mechanism that allows to avoid this undesirable effect. There are three parameters controlling polygon offsets:

- mode: it is possible to turn polygon offsets on and off for certain polygon visualization methods (filled, lines, points) independently;
- factor: when non-zero, this parameter adds more offset for polygons having a larger angle with respect to the viewport;
- units: this value is added to the viewport Z co-ordinates of polygon's pixels;

when negative, it makes pixels closer to the viewport, and otherwise moves pixels away from the viewport.

From now on, in Open CASCADE Technology there are two ways to deal with the problem:

1. AIS_InteractiveObject::SetPolygonOffsets(mode, factor, units) method.

You can move one of the two 3D objects (they should inherit the AIS_InteractiveObject class) a bit away from the eye by turning on polygon offsets for this object:

```
Handle(AIS_Shape) aShape = getAISShape(); // somehow obtain your
interactive object

Handle(AIS_InteractiveContext) aCTX = getAISContext(); // get
interactive context

aShape->SetPolygonOffsets(Aspect_POM_Fill, 1., 1.); // set up polygon
offsets for filled polygons
aCTX->Display(aShape);
```

This sample code makes the OpenGL depth buffer consider aShape presentation as being at a slightly longer distance from the eye than it actually is.

For the second object, polygon offsets can be turned off explicitly:



```
aSecondShape->SetPolygonOffsets(Aspect_POM_Off);
```

The result of this is aSecondShape presentation being visible in the areas of overlapping (and aShape being invisible in such areas).

2. Graphic3d_AspectFillArea3d::SetPolygonOffsets(mode, factor, units) method.

A more flexible and sophisticated way is to control polygon offsets at the level of Graphic3d_Group objects, which are created by the Compute() method of your interactive object. This approach allows to display overlapping primitives belonging to the same object presentation. If you choose this way, you should turn polygon offsets off for Grpaphic3d_Group objects which should have a "priority" in your 3D scene (that is over the other), and for all the others polygon offsets should be on:

```
Prs3d_Root::NewGroup(aPrs3d);
  Handle(Graphic3d_Group) aGrp = Prs3d_Root::CurrentGroup(aPrs3d);
  Handle(Graphic3d_AspectFillArea3d) aFillAsp =
myDrawer->ShadingAspect()->Aspect();
  aFillAsp->SetPolygonOffsets(Aspect_POM_All, 1., 1.);
  aGrp->SetPrimitivesAspect(aFillAsp);
  // here you add graphic primitives to aGrp
  ...
Prs3d_Root::NewGroup(aPrs3d);
  Handle(Graphic3d_Group) aPriorGrp = Prs3d_Root::CurrentGroup(aPrs3d);
  aFillAsp->SetPolygonOffsets(Aspect_POM_Off);
  aPriorGrp->SetPrimitivesAspect(aFillAsp);
  // here you add graphic primitives to aPriorGrp
```

NOTE:

1. Polygon offsets have an impact on polygons only (triangles, quadrangles, etc.), irrespective of the mode they are drawn in (shading, wireframe or points). Such primitives as vertices and lines are not affected by this mechanism.
2. By default, all object presentations have the following polygon offset parameters: mode = Aspect_POM_Fill, factor = 1.0, units = 0.0.

All Graphic3d_Groups have the polygon offset mode Aspect_POM_None by default, so as not to modify polygon offset settings for the corresponding interactive object.



- **Java extractor**

WOK Java extractor wraps Open CASCADE C++ classes into Java classes using JNI (Java Native Interface) calls. It has undergone several corrections and improvements:

- Processing of the Standard_Transient subclasses (i.e. those manipulated by handle) have been corrected. This has been achieved with introducing a new base Java class jcas.Transient (in the jcas.nocdlpack), which extends jcas.Object. The latter remains a base class for all classes manipulated by value as well as for all Standard_Storable subclasses.

- Redundant methods `IsKind()` and `DynamicType()` of former `jcasc.Object` have been removed. This allows correct extracting of classes of the `Standard` package. However, `Standard_Persistent` subclasses still may not be correctly wrapped but this does not limit the extractor use (since these classes are not supposed to be used directly in Java).
 - The extractor now correctly extracts classes without a default constructor. This requirement does not persist anymore. If a null C++ object is passed to Java (what is normally impossible) then an exception is thrown.
 - Processing of large interface units declared in the `jni` unit has been improved. A command line with the list of java files passed to the java compiler now is verified against overflow and is split if necessary.
 - Dependency chain between generated `.java`, `.class`, `.h` and `.cxx` files has been created. Thus, if some classes disappear from the extraction scope (e.g. excluded from the interface unit) their respective generated files will be removed.
 - Generated header `.h` files are now put into the `drv` directory of the unit instead of common `inc` directory of the workbench. This facilitates packaging and keeps the include directory cleaner.
- Various minor Windows-specific improvements:
 - The `WOKSteps_Remove_File` step now actually removes orphan files (e.g. generated from other source files, which have been removed).
 - The `%CMPLRS_CXX_DBMSOpt` parameter is now always used in compilation.
 - Correct start of the WOK session when the `%WOK_ROOTADMDIR%` variable points to a drive different from Open CASCADE installation drive.

Changes

✓ Support of STL streams

- Starting with Open CASCADE 5.1.2 binaries delivered with Open CASCADE on Windows and Linux are built using STL implementation by the supported compilers. The source code can be rebuilt using other implementations (for example STLPort). However Open CASCADE S.A. did not undertake any certification actions on such implementations and therefore cannot guarantee a reliable work with other implementations.
- For most correct support of STL streams the MS Visual C++ 6.0 Studio compiler **must be upgraded** to SP5 (downloadable from www.microsoft.com).

✓ Standard MFC samples

- The OCAF document browser has been integrated to the OCAF MFC sample. By default the Qt-based browser is linked with the sample. The OCAF document browser is a GUI library used for navigation over the document created with Open CASCADE Application Framework (OCAF). The library can be loaded to explore the document at run-time thereby significantly simplifying the debugging process to ensure correctness of the data model and validity of internal document state. The browser is supplied with realization on Fltk, Qt and TclTk graphic toolkits.
- A Progress indicator has been added to the E-Viewer. This Progress indicator is activated during reading of STEP and IGES files.

✓ Data Exchange

- The face orientation fix has been improved to handle cases of wires having touching segments.

✓ Application Framework

- Exception handling has been corrected in class TDocStd_Application that provides methods to store/retrieve documents.
- Implementation of methods: Init, SetValue, Paste and ChangeArray of classes TDataStd_IntegerArray TDataStd_RealArray TDataStd_ExtStrinArrayArray has been changed in order to eliminate extra backups for an attribute in case when Value is not really changed. (see the corresponding improvement).

Warning:

Earlier the ChangeArray() method had the following non-documented behavior: a handle given to the method ChangeArray() was set as an actual attribute data. Therefore it was possible to modify attribute data via this handle after setting a new value by means of the ChangeArray() method.

For instance, the following code could have been used to change an array attribute and fill it with new data:

```
Handle(TDataStd_IntegerArray) anAttr = ...;
Handle(TColStd_HArray1ofInteger) anArr = ...;

anAttr->ChangeArray ( anArr );
```

```
for ( int i=anArr->Lower(); i <= anArr->Upper(); i++ )
    anArr->SetValue ( /* some code */ );
```

Now this code may fail to work because there is no guarantee that anArr given as an argument to ChangeArray() will be the same handle as stored in the attribute.

Therefore all code of this kind in the user applications should be identified and corrected in order to ensure proper work. Either:

- method SetValue() of an attribute should be used,
- or method ChangeArray() should be called after array data are completely prepared.

- **PCDM_ReadWriter_1 class**

The way of creating a relative path to a referenced document on Windows platform has been changed. A possibility has been added to store a referenced document in the network folder.

- **Method TDocStd_XLinkTool::CopywithLink()**

Sets reference to the source document in the target document and gets the document entry. Sets the received document entry to the TDocStd_XLink attribute field.

- The StdResource unit now contains resource files for basic OCAF (the standard file) and XDE (the XCAF file).
- The way of conversion of a double value into a string equivalent in XmlMDataStd_RealDriver.cxx has been changed. Now it does not lose precision at all.

Modeling Algorithms

- **BRepBndLib package**

Function BRepBndLib::Add has been changed. See Reference Documentation for more details.

- **IntWalk_Pwalking class**

The intersection algorithm has been modified to extend an intersection line in the tangent zone. A new sub-algorithm has been added to the Boolean operations algorithm. In order to correctly process some complex cases of source shapes mutual disposition, this new sub-algorithm sews intersection lines and parts of split edges (splits) using their intersection points.

- A new static function IsSweepParallelSpine has been added to GeomFill_Sweep. It checks whether a swept surface is parallel to the spine or not, so the canonic part is defined.
- BRepCheck_Analyzer algorithm has been changed. If SameParameter or SameRange flag of an edge is FALSE this edge is considered invalid now. Presentations on surface of alone edges are not checked now. Such checking is performed only if edges belong to faces. See details in reference documentation (BRepCheck_Analyzer::IsValid).
- The function ComputeSamplePoints of Adaptor3d_TopolTool now works as the same function of BRepTopAdaptor_TopolTool except for the case with torus and definition of the number of U-samples for main non-linear surfaces: it equals 20 instead of 25 in BRepTopAdaptor_TopolTool.



Visualization

- TriangleMeshAdd(TSM_ELEM_DATA d, Tint n, cmn_key *k) function in the OpenGL_tmesh.c file has been corrected: special precautions are now taken if the number of faces to be displayed (data->num_facets value) is 1. This case is treated separately, so as not to run out of the array of face normals while computing vertex normals.
- The Draw_array() function (OpenGL_PrimitiveArray.c) has been improved, so that now uniformly shaded polygons can be safely drawn after the Graphic3d_ArrayOfPolygons containing colored vertices has been drawn, since proper material attributes are restored due to a call to TelResetMaterial() immediately after the GL_COLOR_MATERIAL mode is disabled.
- **V3d_View class methods Reset and FitAll**
A new optional parameter of type Standard_Boolean has been added to enable the defer view update if necessary.
- **Visual3d_view::Redraw()** methods have been corrected so as to properly set the Z buffer state (on/off).

Installation

- Starting with Open CASCADE 5.1.1:
 - debug versions of libraries (dll and lib) and executables (exe) for Windows are no longer delivered;
 - dynamic-link libraries (dll) and executables (exe) for Windows are located in ros/win32/bin (this will allow to shorten the Open CASCADE directory list added into the PATH system variable);

Building Tools

- Dynamic-link libraries (dll) and executable (exe) files built by MS Visual Studio projects on the Windows platform, are now placed to the ros/win32/bin folder and replace their corresponding original libraries and executable files, respectively binaries for the debug mode are placed in the ros/win32/bind folder.
- New options for the GNU autoconf/automake tools to enable STLPort use (see "Support of STL streams" above).

Test Harness Draw

- Using Draw Test Harness in the form of a single executable (DRAWEXE) with extensible command libraries (dynamically loaded as plugins) has been implemented. Old executables (AISViewer, TTOPOLOGY and etc.) have been deleted.

See [Appendix 3](#) for details on Test Harness changes.



- DRAW command "checkshape" has been changed. DRAW commands "checkbrep" and "checktopshape" have been removed. See the new syntax of "checkshape" command in documentation (paragraph 5.1.5 of Test Harness User Guide).

Qt Samples

- Qt Import/Export and Visualization Performance Meter samples have been ported to Qt 3.2.1. However, the delivered binaries remain to be linked to Qt 2.3.0.

Porting

- Open CASCADE Technology has been ported to Microsoft Visual C++ 7.1 compiler. The possibility to build Open CASCADE with VC7.1 and correct operation of achieved libraries have been tested and checked during the porting of Open CASCADE to VC7.1.

However, binaries delivered with this Version 5.2 of Open CASCADE Technology have been built with Microsoft Visual C++ 6.0.

- Open CASCADE has been ported to Linux RedHat 8.0 with support of gcc 3.2. Binaries delivered starting with Open CASCADE 5.1.1 have been built using this Linux distribution and the compiler.
- Starting with Open CASCADE 5.1.2, Windows® XP Professional has been included on the list of supported platforms.

Bug Fixes



- Open CASCADE 5.2 incorporates **172** modifications (bug fixes, enhancements and other corrections) over version 5.1. For details, refer to [Appendix 1](#).

Appendix 1: Open CASCADE 5.2 Bug Fixes

- [Data Exchange](#)
- [Foundation Classes](#)
- [Modeling Algorithms](#)
- [Modeling Data](#)
- [Application Framework](#)
- [Visualization](#)
- [Shape Healing](#)
- [WOK](#)
- [Test Harness](#)

Data Exchange, 38 bug fixes	
ID	Short Description
210	Improve FixShape to correct case of touching wires
572	Written DXF file can not be read by Volo View Express.
1080	bug in step writer
2368	Invalid result after reading a STEP file.
2513	Class MoniShapeOperAdv belongs to two toolkits
2821	IGES file is not imported correctly
3004	Regressions after improvement performance
3142	Case igs/005/A2 has not been finished everytime. Regress.
3147	Problems of reading IGES file.
3397	Incorrect work of ShapeFix_ComposeShell
3430	Incorrecting creation pcurves
3925	Exception during reading file using XDEDRAWEXE
3926	Exception during reading a file using XDEDRAWEXE
3962	Can't read stp file
3997	No check of pointer is made for the null value
4082	Problems during translation of IGES files
4142	Regression on reading e3i files
4503	Integration of perf meter
4504	Integration of perf meter
4562	Problems with compilation SMDS package
4648	Problems with writing to STEP
4792	Shape become invalid after Canonical Recognition (no shading no face)
4819	Integration changes for optimization of XDE reading
4822	Translation surfaces of revolution and extrusion instead of canonical
4968	ShapeHealing can not correct invalid self-intersecting wire
5027	Incorrect result translating wire with loops from STEP
5089	Progress Indicator works wrong during the translation of assemblies from STEP.
5130	Can not read STEP file with curves bounded by points
5187	XCAFDoc_DocumentTool::IsXCADFDocument is declared but not implemented
5192	After translation the resulting curves are incorrect in a STEP file.



5237	Progress Indicator inserted in the XSControl_TransferRoots
5303	Integration of changing STEP translator after CAX-IF TR13J
5708	New feature for fixing the position of vertices
5881	Incorrect reading of tolerance from an IGES file
5975	Regressions on STEP translator
5989	Exception during reading an attached file.
6053	Regression on writing external references in the STEP AP203.
6157	Patch on the reading STEPAP209
Foundation Classes, 9 bug fixes	
ID	Short Description
3277	Memory leak in TCollection strings
4040	The method NCollection_DataMap::Bind must change the value anyway
4146	Various improvements in NCollection classes
4148	The work of Standard allocator when MMGT_OPT=0 has been improved
4417	OSD_Error incorrectly treats some Windows error codes
5091	NCollection classes should raise exceptions according to OCC standard
5092	OSD_FileNode: methods AccessMoment and CreationMoment fail on Win9x
5600	Corrupted header file NCollection_Stack.hxx
6046	Memory leak in math_StackManager
Modeling Algorithms, 74 bug fixes	
ID	Short Description
311	Hung up and incorrect result in Segment (2d and 3d)
435	Exception Standard_ConstructionError is raised in the GeomConvert_CompCurveToBSplineCurve
479	GeomPlate_Surface::Uiso and GeomPlate_Surface::Viso return a NULL Geom_Curve
496	Exception occurs during fuse operation
497	Results of cut and fuse operations are faulty shapes although arguments are valid.
519	Draw hungs up in BRepAlgoAPI_Cut function during loading the e3i-file under pkv-GalleryTest workbench
523	Draw "hangs up" in BRepAlgoAPI_Cut function during loading the e3i-file under pkv-GalleryTest workbench
528	Result of bcut command is faulty although arguments are valid
567	Can not intersect two Rectangular Trimmed Surfaces
578	Bug in BRepClass3d_SolidClassifier
583	Result of bopcut operation is wrong
584	Result of cut operation is wrong
600	Result of BOPCOMMON operation is a non-closed shape in spite of the fact that source solids are valid
605	No faces in result of the PIPE command
620	No possibility to cut the shapes.
629	Exception when attempting to create a solid by command PIPE
763	Bad result of fuse operation between two cylinders
943	Wrong projection of 3dcurve on surface.
1077	A bug in boolean operations
1477	Problems in MakePipeShell
1764	Unstable work of checkshape for detection of cases of self-intersection wire



2500	BRepAlgoAPI_Section has a problem with calculation
2755	Boolean Bug in OCC5.0 with the shape having an ellipse curve
2991	BrepMesh_IncrementalMesh takes forever to mesh some faces
3403	A little bug in XCAFDoc_DocumentTool.cxx
3502	Incorrect work of command sprogs
3565	Dynamic loading of Draw Commands
3565	Dynamic loading of Draw Commands
3643	Moving the tool during the cut operation fails
3644	Moving a component of the intersection operation fails
3645	Moving a component of the intersection operation fails
3727	Problem of Geom2dAPI_InterCurveCurve
3896	BRepCheck_Analyzer does not detect faulty faces in a solid shape.
3918	Exception during retrieval
4315	Invalid result of intersection of 2d curves
4315	Invalid result of intersection of 2d curves
4426	Incorrect result of intersection in 2D between circle and line
4455	Invalid result or exception in offset algorithm
4717	A regression of new Boolean Operations
4900	Incorrect work of DRAW command bfuse
4993	Problem in boolean fusion
4994	Problem in boolean fusion
5067	Regressions on windows XP. Pipes
5073	Problem with sweep
5093	Wrong result of the pipe algorithm
5110	Cut operation throws an exception
5112	Cut operation produces invalid shape
5113	Cut and common operations produce invalid results
5246	Bug of new Boolean Operations.
5335	Error in MakePipeShell: program hangs up
5336	Incorrect behavior of MakePipeShell algorithm
5576	Improvement for BRepExtrema_DistShapeShape
5579	Providing correct behavior of exception catching code both STL and non-STL streams
5581	Impossible to make an offset on a shape
5607	Porting of Local Operation algorithms to use New Boolean operations
5634	Porting of 3d Offset and Features algorithms to use New Boolean operations
5645	Useless messages in DRAW while performing command "mkoffset"
5654	Improvement of Surface/Surface Intersection algorithm
5691	Exception is raised from Triangle/triangle.c : triangulate()
5805	BRepOffsetAPI_MakeThickSolid behaves unstably and produces very different results.
5865	Bug in SplitShape when splitting through seam edge
5867	Incorrect initialization of BRepOffsetAPI_NormalProjection
5882	Intersection is not found between line and torus
5883	Error of comparison in BRepTopAdaptor_FClass2d
5898	Porting of OCC to VC7
5918	After first unsuccessful call of BRepAlgoAPI_Section the second one leads to an exception
5931	Substitution of the existing set of valuation DRAW commands by one (checkshape)
6001	Cannot intersect a closed composite curve with a plane
6002	ThruSections produces a bad shape if edges have locations
6012	Problem with performance of new BOP on a complex bspline surface



6068	Cyclic dependency between toolkits in DEB mode
6129	Problem of constructing a face from its four bounding edges using the <code>GeomFill_ConstrainedFilling</code> package
6183	Incorrect result of boolean fuse
6226	Handling of touching cases based on canonical surfaces by a boolean operation
Modeling Data, 3 bug fixes	
ID	Short Description
3721	When using Boolean Operations an exception is raised for some tolerance values
5739	<code>GCPnts_UniformAbscissa</code> gives an incorrect distribution of points
6020	Forgotten Location during Edge reading from <code>.brep</code> file
Application Framework, 26 bug fixes	
ID	Short Description
2933	In Std and XML plugins, and in PCDM messaging to cout has been removed using the <code>CDM_MessageDriver</code>
3427	New <code>ExtStringArray</code> attribute - improvement
3548	Multi Trans. Manager: need to know if just committed transaction had changes
3641	The result of moving during the cut operation fails
3646	Moving a component during the intersection operation fails
3713	<code>MultiTransactionManager</code> : bug in treatment of undo limit
3960	Undo does not work after copying an array attribute into another one
3985	XML persistent incorrect process (write/read) unicode strings
4149	Clear relocation table at the end of read/write a document using XML persistence
4150	Some improvements in the LDOM package
4327	Incorrect exception handling in <code>TDocStd_Application</code>
4800	Incorrect processing of Plugin file absence
5023	Performance regression in opening OCAF file
5031	Improve the performance of <code>TDF_AttributeIterator</code>
5052	Saving document to a file: dot sign is attached to a file name.
5053	Implement methods to access document saved time.
5559	It is impossible to explicitly clear driver tables
5712	Incorrect creation of the Messages for Message Driver
5715	XLink update fails
5716	Incorrect computation of the network path during XLink update.
5731	Space character in file name are removed
6007	Completion of binary OCAF persistence
6010	A value of <code>TDataStd_RealArray</code> saved in XML format is rounded to about 4 digits
6097	Conversion of double value to string is performed with loose of precision in <code>XmlMDataStd_RealDriver</code>
6098	Display mode is absent in XML persistence of OCAF
6139	Removing API redundant
Visualization, 14 bug fixes	
ID	Short Description



230	Numeric Error occurs in V2d_View::WindowFit() , Magnify()
2934	Optimized performance on 3D text output in 3D viewer.
4147	A possibility of conditional update has been added in methods FitAll and Reset of V3d_View
4201	Performance problem with SelectMgr
4298	Memory beyond array bound read in TriangleMeshAdd() (OpenGL_tmesh.c)
4373	Improper analysis of current display mode in AIS_InteractiveContext methods
4692	It is impossible to highlight Graphic3d_ArrayOfPolylines primitives.
4723	Material attributes are wrong after drawing an array of polygons+vertex colors.
4844	Function V3d_View::Rotate uses static variables
4895	Eliminate Z buffer artifacts when some overlapped shaded polygons are drawn
5090	Advanced mechanism for highlighting selected/detected entity owners
5120	Impossible to visualize shapes from the XDE documents in Draw on Windows
5701	Visual3d_View::Redraw(): Depth buffer is not correctly set up
6080	Integration of Mesh Visualisation Service (MeshVS) component
Shape Healing, 1 bug fix	
ID	Short Description
2851	Regression on performance
WOK, 5 bug fixes	
ID	Short Description
563	Added option "-instances=static" to the compilation command line
4062	The content of development version of WOK does not correspond to release one.
4732	Avoiding of compilation errors (jcas package).
5082	Some corrections and improvements in WOK Java extractor.
5083	Some Windows-specific corrections and improvements in WOK
Test Harness - Draw, 2 bug fixes	
ID	Short Description
3565	Dynamic loading of Draw Commands
5601	Multiple run of pload command leads to memory problems



Appendix 2: list of removed entities

CDF package:

Removed classes:

CDF_Selection
 CDF_SelectionIterator
 CDF_ClipBoard
 CDF_CheckDocumentToStore

Removed enumeration: SelectionStatus

Class CDF_Session:

Removed redundant methods:

Clipboard()
 SetClipboard()
 CanStore()
 CanWrite()
 CanCopyToEuclid3()
 CopyToEuclid3()
 CanCopy()
 Copy()
 CanPaste()
 Paste()
 CanPastewithLink()
 PastewithLink()
 CanRemove()
 Remove()
 CanCut()
 Cut()
 CanInsert()
 InsertwithRefrence()
 LoadDiver(4 parameters)
 LoadDriver(5 parameters)

Removed a field of the CDF_ClipBoard type

Class CDF_Application:

Removed redundant methods:

CanStore()
 Store()
 CanCopy()
 Copy()
 CanPaste()
 Paste()
 CanCut()
 Cut()
 CanRemove()
 Remove()
 CanWrite() – set of methods
 Write() – set of methods
 CanRead() – set of methods
 Read() – set of methods
 CanClose()
 Close()
 CanInsert()
 Insert()
 CanNew()
 New()
 Findwriter() - set of methods
 writer() - set of methods

Class CDF_Store:

Removed friend class

CDF_CheckDocumentToStore

Class CDF_MetaDataDriverFactory:

Removed redundant methods:

```
virtual Handle_CDF_MetaDataDriver Build(
    const TCollection_ExtendedString& DMEnginePath,
    const TCollection_ExtendedString& Host,
    const TCollection_ExtendedString& Port,
    const TCollection_ExtendedString& EDMSchemaPath) const;
virtual Handle_CDF_MetaDataDriver Build(
    const TCollection_ExtendedString& DMEnginePath,
    const TCollection_ExtendedString& Host,
    const TCollection_ExtendedString& TCPIP,
    const TCollection_ExtendedString& Unix,
    const TCollection_ExtendedString& EDMSchemaPath) const;
```

Class FWOSPlugin_DriverFactory:

Removed redundant methods:

```
virtual Handle_CDF_MetaDataDriver Build(
    const TCollection_ExtendedString& DMEnginePath,
    const TCollection_ExtendedString& Host,
    const TCollection_ExtendedString& Port,
    const TCollection_ExtendedString& EDMSchemaPath) const;
virtual Handle_CDF_MetaDataDriver Build(
    const TCollection_ExtendedString& DMEnginePath,
    const TCollection_ExtendedString& Host,
    const TCollection_ExtendedString& TCPIP,
    const TCollection_ExtendedString& Unix,
    const TCollection_ExtendedString& EDMSchemaPath) const;
```



Appendix 3: Test Harness Draw (changes)

Since version 5.1.1 Open CASCADE introduces a single executable in the DRAW Test Harness that supersedes the several separate executables that existed before. Respectively the user does not need to have his own executables to activate his custom commands. All he needs to do is to implement the commands themselves, they will be activated in the common executable. This executable is now called **DRAWEXE**.

Commands grouped in toolkits can now be loaded at run-time thereby implementing dynamically loaded plug-ins. Thus, the user can work only with those commands that suit his needs adding these commands dynamically without leaving the Test Harness session.

Declaration of available plug-ins is done through the special resource file(s). The **pload** command loads the plug-in in accordance with the specified resource file and activates the commands implemented in the plug-in. The whole process of using new advantages of the plug-in mechanism as well as instructions for extending Test Harness are described below.

Launching DRAW Test Harness

Test Harness executable DRAWEXE is located in the \$CASROOT/<platform>/bin directory (where <platform> is win32 for Windows, SunOS for Sun Solaris and Linux for Linux operating systems). Prior to launch it is important to make sure the environment is correctly set-up (usually this is done automatically after the installation process on Windows or after launching specific scripts on Unix/Linux) - refer to Technical Documentation for details.

Plug-in resource file

Open CASCADE is shipped with the DrawPlugin resource file located in the \$CASROOT/src/DrawResources directory.

The format of the file is compliant with standard Open CASCADE resource files (see the Resource_Manager.cdl file for details).

Each key defines a sequence of either further (nested) keys or a name of the dynamic library. Keys can be nested down to an arbitrary level. However, cyclic dependencies between the keys are not checked.

Example (excerpt from DrawPlugin):

```
OCAF           : VISUALIZATION, OCAFKERNEL
VISUALIZATION  : AISV
OCAFKERNEL     : DCAF

DCAF           : TKDCAF
AISV           : TKViewerTest
```

Activation of commands implemented in the plug-in

To load a plug-in declared in the resource file and to activate the commands the following command must be used in Test Harness:

pload [-PluginFileName] [[Key1] [Key2]...], where:





<-PluginFileName> Defines the name of a plug-in resource file (prefix "-" is mandatory) described above.

If this parameter is omitted then the default name DrawPlugin is used.

<Key>... Defines the key(s) enumerating plug-ins to be loaded.

If no keys are specified then the key named DEFAULT is used (if there is no such key in the file then no plug-ins are loaded).

According to the Open CASCADE resource file management rules, to access the resource file the environment variable CSF_<PluginFileName>Defaults (and optionally CSF_<PluginFileName>UserDefaults) must be set and point to the directory storing the resource file. If it is omitted then the plug-in resource file will be searched in the \$CASROOT/src/DrawResources directory.

Examples:

Draw[]> pload -DrawPlugin OCAF

Will search the resource file DrawPlugin using variable CSF_DrawPluginDefaults (and CSF_DrawPluginUserDefaults) and will start with the OCAF key. Since the DrawPlugin is the file shipped with Open CASCADE it will be found in the \$CASROOT/src/DrawResources directory (unless this location is redefined by user's variables). The OCAF key will be recursively extracted into two toolkits/plugin-ins: TKDCAF and TKViewerTest (e.g. on Windows they correspond to TKDCAF.dll and TKViewerTest.dll). Thus, commands implemented for Visualization and OCAF will be loaded and activated in Test Harness.

Draw[]> pload (equivalent to pload -DrawPlugin DEFAULT).

Will find the default DrawPlugin file and the DEFAULT key. The latter finally maps to the TKTopTest toolkit which implements basic modeling commands.

Mapping between former separate Test Harness executables and the new plug-ins

Before version 5.1.1 Open CASCADE used to be shipped with several separate executables providing different sets of commands. The following table represents the mapping between former executables and new plug-ins.

Former executable	Current key
AISViewer	VISUALIZATION
TCAF	OCAF
TTOPOLOGY	MODELING
XDEDRAWEXE	DATAEXCHANGE
XSDRAWEXE	DATAEXCHANGEKERNEL

For instance, in order to activate commands available in the former AISViewer executable, now it is enough to use the command pload VISUALIZATION.

Extending Test Harness with custom commands

The following chapters explain how to extend Test Harness with custom commands and how to activate them using a plug-in mechanism.

Custom command implementation

Custom command implementation has not undergone any changes since the introduction of the plug-in mechanism. The syntax of every command should still be like in the following example.



Example:

```
static Standard_Integer myadvcurve( Draw_Interpreter& di,
Standard_Integer n,
char** a)
{
...
}
```

For examples of existing commands refer to Open CASCADE (e.g. GeomliteTest.cxx).

Registration of commands in Test Harness

To become available in the Test Harness the custom command must be registered in it. This should be done as follows.

Example:

```
void MyPack::CurveCommands(Draw_Interpreter& theCommands)
{
...
char* g = "Advanced curves creation";

theCommands.Add ( "myadvcurve", "myadvcurve name p1 p2 p3 - Creates my advanced
curve from points",
__FILE__, myadvcurve, g);
...
}
```

Creating a toolkit (library) as a plug-in

All custom commands are compiled and linked into a dynamic library (.dll on Windows, or .so on Unix/Linux). To make Test Harness recognize it as a plug-in it must respect certain conventions. Namely, it must export function `PLUGINFACTORY()` accepting the Test Harness interpreter object (`Draw_Interpreter`). This function will be called when the library is dynamically loaded during the Test Harness session.

This exported function `PLUGINFACTORY()` must be implemented only once per library.

For convenience the `DPLUGIN` macro (defined in the `Draw_PluginMacro.hxx` file) has been provided. It implements the `PLUGINFACTORY()` function as a call to the `<Package>::Factory()` method and accepts `<Package>` as an argument. Respectively, this `<Package>::Factory()` method must be implemented in the library and activate all implemented commands.

Example:

```
#include <Draw_PluginMacro.hxx>

void MyPack::Factory(Draw_Interpreter& theDI)
{
...
//
MyPack::CurveCommands(theDI);
...
}

// Declare entry point PLUGINFACTORY
DPLUGIN(MyPack)
```



Creation of the plug-in resource file

As mentioned above, the plug-in resource file must be compliant with Open CASCADE requirements (see Resource_Manager.cdl file for details). In particular, it should contain keys separated from their values by a colon (":").

For every created plug-in there must be a key. For better readability and comprehension it is recommended to have some meaningful name.

Thus, the resource file must contain a line mapping this name (key) to the library name. The latter should be without file extension (.dll on Windows, .so on Unix/Linux) and without the "lib" prefix on Unix/Linux.

For several plug-ins one resource file can be created. In such case, keys denoting plug-ins can be combined into groups, these groups - into their groups and so on (thereby creating some hierarchy). Any new parent key must have its value as a sequence of child keys separated by spaces, tabs or commas. Keys should form a tree without cyclic dependencies.

Examples (file MyDrawPlugin):

```
! Hierarchy of plug-ins
ALL           : ADVMODELING, MESHING
DEFAULT      : MESHING
ADVMODELING  : ADVSURF, ADVCURV

! Mapping from naming to toolkits (libraries)
ADVSURF      : TKMyAdvSurf
ADVCURV      : TKMyAdvCurv
MESHING      : TKMyMesh
```

For other examples of the plug-in resource file refer to the **"Plug-in resource file"** chapter above or to the \$CASROOT/src/DrawPlugin file shipped with Open CASCADE.

Dynamic loading and activation

Loading a plug-in and activating its commands is described in the **"Activation of the commands implemented in the plug-in"** chapter.

The procedure consists in defining the system variables and using the pload commands in the Test Harness session.

Example:

```
Draw[]> set env(CSF_MyDrawPluginDefaults) /users/test
Draw[]> pload -MyDrawPlugin ALL
```

Appendix 4: New Test Harness Draw commands

- A set of new commands has been added to Test Harness Draw:
 - meshfromstl – creates a MeshVS_Mesh object based on STL file data. The object will be displayed immediately.
 - meshdispmode – changes the display mode of object "objectname". The displaymode is integer, which can be 1 (for wireframe), 2 (for shading mode) or 3 (for shrink mode).
 - meshselmode – changes the selection mode of object "objectname". The selectionmode is integer OR-combination of mode flags. The base flags are following:



- 1 – node selection,
- 2 – 0D elements (not supported in STL)
- 4 – links (not supported in STL)
- 8 – faces

- meshshadcolor - changes the face interior color of object "objectname". The red, green and blue are real values between 0 and 1.
- meshlinkcolor - changes the color of face borders for object "objectname". The red, green and blue are real values between 0 and 1.
- meshmat - changes the material of object "objectname". Material is represented with an integer value as follows (equivalent to enumeration Graphic3d_NameOfMaterial):

0 – BRASS,	11 - SATIN,
1 – BRONZE,	12 - METALIZED,
2 - COPPER,	13 - NEON_GNC,
3 - GOLD,	14 - CHROME,
4 - PEWTER,	15 - ALUMINIUM,
5 - PLASTER,	16 - OBSIDIAN,
6 - PLASTIC,	17 - NEON_PHC,
7 - SILVER,	18 - JADE,
8 - STEEL,	19 - DEFAULT,
9 - STONE,	20 - UserDefined
10 - SHINY_PLASTIC,	
- meshshrcoeff - changes the value of shrink coefficient used in shrink mode. In the shrink mode the face is shown as a congruent part of a usual face, so that shrink coefficient controls the value of this part. The coeff is a positive real number.
- meshshow - displays mesh in the viewer (if it is erased).
- meshhide - hides mesh in the viewer.
- meshhideSel - hides only selected entities. The other part of mesh remains visible.
- meshshowSel - shows only selected entities. The other part of mesh becomes invisible.
- meshshowAll - changes the state of all mesh entities to visible.