# SALOME 3.2.0b1 development release announcement

## General information

OPEN CASCADE is pleased to announce SALOME 3.2.0beta (b1) minor development version. It is an intermediate development release that contains the results of planned improvements and bug fixes with reference to the SALOME 3.2.0a2 development release. SALOME 3.2.0b1 passed a complete GUI test campaign.

The major goal of SALOME 3.2.0b1 is to provide qualified latest state before final version and to share the current development state among CEA/EDF/OPEN CASCADE. This document describes the main differences since SALOME 3.2.0b1 version.

Since the 3.2.0b1 is a development release, this document mainly addresses to the developers.

# Major improvements according to SALOME 3.2.0 alpha 2

## Installation procedure

- From version 3.2.0b1 the installation procedure generates an additional folder in the user's home directory. This folder includes scripts which allow launching SALOME servers in different modes according to the Application notion introduced by EDF. The name of this folder is generated according to the number of the installed SALOME version: for example, for version 3.2.0b1 the folder name is salome_appli_3.2.0b1. The environment script which is required for SALOME launching is also automatically generated in this folder by the installation procedure.
  NOTE: this folder is created only if the user installs KERNEL module binaries package.
- Debian installation procedure forces installation of all products from binaries as a default option.
- Unnecessary products (happy doc)  have been removed from install proc
- New version of doxygen is included in the installation package (1.4.6).

## General modifications

- Salome has been ported to Mandriva. As Mandriva is a 64 bit platform, SALOME VTK functionality has been ported to VTK 4.4. Consequently, in order to support both versions of VTK (current 4.2.6 and new 4.4) we need to use vtkFloatingPointType instead of bare float point type for all VTK functionalities.
- Context help has been added in most dialog boxes in all modules.

# KERNEL

SALOME_Session.idl has been moved from GUI_SRC to KERNEL_SRC as it modifies the SALOME CORBA module.

### Configuration procedure

build_configure does not need anymore to be invoked from the source directory.
**A lot of no more relevant files have been removed from sources**

### SALOME Application

Improvement of cleaning process when the application is killed.

### Launch procedure

- SalomeApp.xml is now split among all modules (see GUI modifications for more details);
- runSalome support “—modules” option has been restored.

### Developer and integrator documentation (in source tarball)

doc directory contains several documents in progress:

- SALOME Application : how to install and use the SALOME application on several computers;
- Unit Tests : organization of sources, unit tests procedures;
- Kernel Resources: some tools for developers (other documents are obsolete).

User TUI documentation updated.
The possibility to generate documentation from Python source by DOXYGEN has been added.

### SALOME_Comm package (efficient distributed memory data transfer)

A matrix interface has been added.

### CORBA transfer service for large files

There is a C++ class (SALOME_FileTransferCORBA) which provides a transfer service for copies of large files, given a distant computer hostname and a file path on this computer. This service relies on container servers (LifeCycleCORBA). The service keeps trace of copies already edited on a computer, to avoid multiple copies (and if the reference file is on the local computer, no copies are created). It is also possible to share an object representing the file between CORBA servers and client, to obtain local copies.
The interface is available in Python via Swig.

### Unification of Python scripts used either inside GUI server or outside it (batch mode)

A SALOME Session can be launched with or without GUI (with a Python interpreter interface only, see doc/SALOME_Application) and the Python interpreter can be launched outside an already running SALOME session and connected to it.
Provided a Python script does not use Swig interface from GUI, it can be used inside or outside GUI server. For instance, scripts based on geompy.py or smesh.py work with external interpreters.
Consequently the non regression test base has been updated to use geom.py and visu.py, SMESH tests are in progress.

### Unit Tests

Python Unit testing capability has been added, it tests application written on LifeCycleCORBA, and also tests Python Swig interface.

# VISU module

Some improvements have been introduced in this module:

- Addition of preferences in VISU module: now general preferences contain preferences for VISU presentations (colors, type of presentation etc)
- Toolbar buttons to import/export MED files have been added in the module
- Structured mesh elements in VISU are now supported!
- Generation of AVI animations: the user can also choose generation of animation during generation of pictures (the availability of this functionality depends on existence of mpeg tools in Linux)

# SMESH module

### GUI for creation of mesh and algorithm dependency

In order to reflect dependencies between algorithms and between the algorithm and hypotheses, the dialogue for creation/edition of meshes/sub meshes has been improved.

As before, the dialogue contains 3 tabs, each defining hypotheses of a certain dimension. Each tab has

- a list of available algorithms (combobox);
- two lists of existing hypotheses (comboboxes): for main and additional hypotheses;
- two "Create hypothesis" buttons, which, when pressed, show a list of available hypotheses;
- two "Edit hypothesis" buttons invoking a dialogue for editing of the hypothesis selected in the list of existing hypotheses.

The algorithm - algorithm dependencies are reflected as follows.
When an available algorithm type is selected, the lists of algorithms in the other tabs are updated:

- lists of available algorithms change to contain only the algorithms compatible with the selected one;
- if a pre-selected algorithm is incompatible with the selected one, it is deselected;
- if there is only one compatible algorithm in the list, it is selected;
- if no algorithms are selected in a tab, the other tabs contain all available algorithms.

The algorithm - hypotheses dependencies are reflected similarly.
When an algorithm is selected, the lists of available and existing hypotheses in each tab are updated to contain only hypotheses compatible with the algorithm of the tab, selected by the user or automatically:

- if there exists only one compatible hypothesis in the list, and it is not optional, it is selected;
- if no algorithms are selected, then the lists of hypotheses are empty.

The order of tabs has been changed: the tab for 3D Hypotheses comes first. As currently, for most 3D algorithms (except GHS3D), there is only one compatible 2D and 1D algorithm, when a 3D algorithm is selected, all other algorithms are selected automatically.
The tab of highest accessible dimension always contains all types of algorithms of this dimension.
All GUI controls providing selection from a list are disabled when the list is empty.

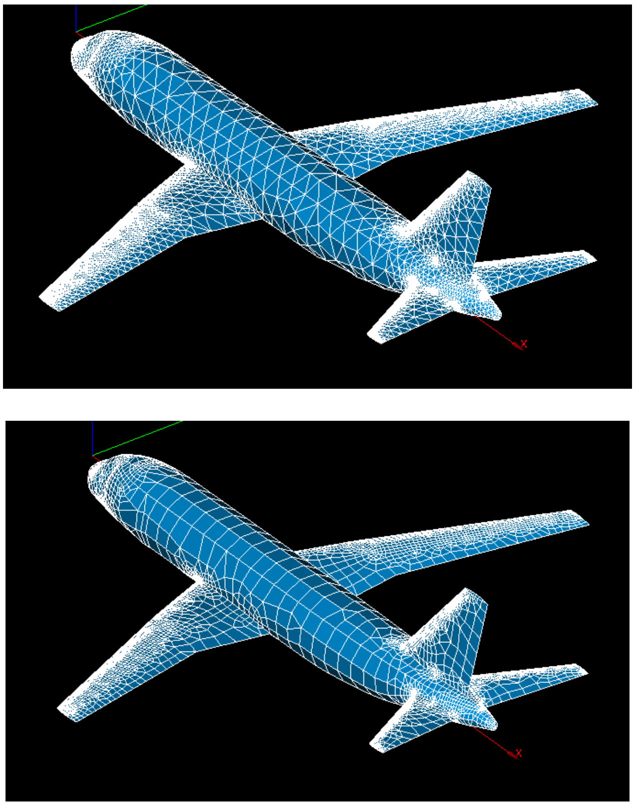
## NETGEN 2D-3D plugin

The study on NETGENPLUGIN porting resulted in the adoption of a new version of Netgen 4.5 (before that we used Netgen 4.3).
NOTE: The version of Netgen can change to 4.4 in the final SALOME 3.2 version.
It includes

- Porting NETGENPLUGIN module to Netgen ver.4.5.
- Adding two new independent algorithms to generate 2D and 3D meshes directly from CASCADE shape.
- Adding two new hypotheses for new algorithms.
- Adding a GUI library providing a dialog to edit hypotheses

The following minor modifications have been introduced into SMESH module:

- Specific icon files for Netgen plugin to be displayed in the object browser have been into the resources of SMESH module.
- SMESH_SWIG/smesh.py file has been modified to support new Netgen algorithms in this python module.
- Two new python test scripts: SMESH_mechanic_netgen.py and SMESH_fixation_netgen.py has been added into SMESH_SWIG directory. The first generates a mixed quadrangle/triangle 2D mesh. The second generates a tetrahedral 3D mesh.



“Flight.brep” model which failed with default parameters on
MEFISTO algorithm is now computed perfectly

## Plugin modifications

To be able to import module of SMESH plugin into the script produced by DumpPython functionality, the following method have been added into GenericHypothesisCreator_i class (SMESH_SRC/src/SMESH_I/SMESH_Hypothesis_i.hxx), which is a root of classes creating hypotheses in plugins:

```
class GenericHypothesisCreator_i
{
public:
  ...
  // return the name of IDL module
  virtual std::string GetModuleName() = 0;
};
```

This method returns the name of python module defining classes of hypotheses created by a plugin; this module is generated from IDL file of the plugin.

The method is defined in, for example, NETGENPLUGIN as follows (NETGENPLUGIN_SRC/src/NETGENPlugin/NETGENPlugin_i.cxx)

```
template <class T> class NETGENPlugin_Creator_i:public
HypothesisCreator_i<T>
{
  // as we have 'module NETGENPlugin' in NETGENPlugin_Algorithm.idl
  virtual std::string GetModuleName() { return "NETGENPlugin"; }
};
```

As a result, DumpPython script contains lines importing modules of plugins which have created some hypotheses:

```
def RebuildData(theStudy):
        ...
    import StdMeshers
    import NETGENPlugin
```

## Update of MEFISTO algorithm

StdMeshers package has been updated by the latest version of MEFISTO algorithm. In some case it works better, but sometimes it produces bad meshes.

## UNV import-export improvements

From now on, the UNV interface supports reading-writing of groups.

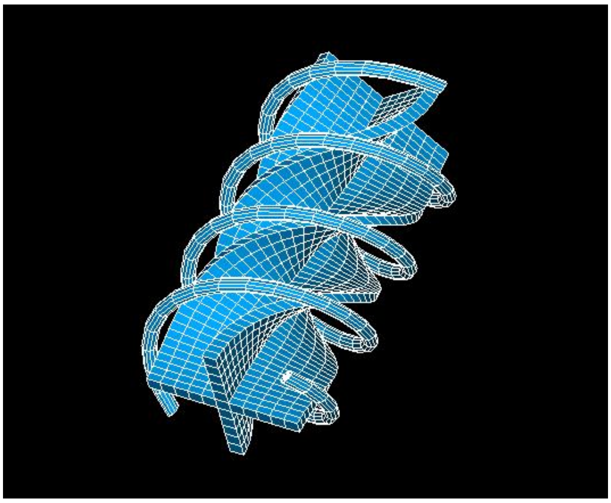
## Quadratic mesh improvements

- Advanced mesh info has been updated to show the number of quadratic elements
- Special functions allowing to convert  quadratic elements to usual first order elements and visa versa have been added in SMESH module

## New IDL functionality

New functions have been added to SMESH. They provide the possibility to work with mesh without GEOM data. For more details please refer to IDL documentation.

```
double_array GetNodeXYZ(in long id);
long_array GetNodeInverseElements(in long id);
long GetShapeID(in long id);
long GetElemNbNodes(in long id);
long GetElemNode(in long id, in long index);
boolean IsMediumNode(in long ide, in long idn);
long ElemNbEdges(in long id);
long ElemNbFaces(in long id);
boolean IsPoly(in long id);
boolean IsQuadratic(in long id);
double BaryCenter(in long id);
```



Mesh generated by a script without geometry
(the idea is taken from GMESH web site)

# GEOM module

GUI access has been added in GEOM to check and correct compound of blocks functionality. It can be accessed via Measures -> Check Compound of Blocks.

# GUI module

## Resource files modification

- From now on, we have SalomeApp.xml configuration file not only in GUI module - each SALOME component (except mesh plug-ins) has its own copy of this file. So, if you add or modify some preferences for a certain component, please, put them in the corresponding SalomeApp.xml file. It is not allowed to put such component-related stuff in the general SalomeApp.xml (which is in GUI module). Those who develop custom SALOME-based modules, have to create their own SalomeApp.xml files and put all preferences there.

- In addition, user preference files from the nearest version (.SalomeApprc_version_name) are imported automatically on the first run of this version of SALOME.

## SIGFPE problem on some graphic card configuration (standard NVIDEA drivers on Mandrake 10.1)

Actually the problem existed before version 3.2.0a2 but it was not visible, since the FPE signals were not raised. The matter is that recently the OCC function OSD::SetSignal(Standard_Boolean aFloatingSignal) has been corrected so that it could raise FPE signal on Linux when an arithmetic exception occurs (the system function feenableexcept is used there).
To have this bug fixed we have updated the call to OSD::SetSignal in SALOME code so that FPE signals were disabled unless Salome is compiled in debug mode and the environment variable DISABLE_FPE is not set to 1.
In other words, in optimized mode FPE are not raised, as it has been in earlier versions, and in debug mode the code decides to raise or not FPE signals depending on the value of the variable DISABLE_FPE.

## Splash logo modifications

Note that the processing of the splash screen window has been modified in  SALOME:

1) A new class QtxSplash which presents the splash screen window similar to the Qt's QSplashScreen class (the latter does not suit all SALOME requirements) has been implemented.
2) The main difference from the previously used approach is that this widget can be shown even if the main application event loop has not yet been started, which allows displaying splash at the very first step.
3) Correspondingly SALOME_Session_Server has been improved to use this class instead of the previous one (InquireServersQThread.* files have been removed from the compilation process since they are no more needed).
4) Splash screen support has been implemented for the light SALOME configuration - the SUITApp package has

been modified correspondingly.

5) Splash screen is now much more customizable than earlier (in both configurations - full and light). It is now possible to customize the following parameters of splash screen window:
    a. splash screen picture (file);
    b. progress bar color(s) (one color or two colors for gradient progress bar, vertical or horizontal type of gradient);
    c. splash screen info text (any text can be displayed in the splash window; there are also some predefined templates support: %A for application name, %V for application version, %L for license info and %C for copyright).
    d. splash screen text colors (one or two colors, second one is used for text shadow).

The parameters of splash screen are customizable via configuration files. Default values are already provided in the corresponding files: LightApp.xml for the light configuration and SalomeApp.xml for the full version of SALOME.

### Small ergonomic improvement

- If Mozilla browser does not exist, a special dialog gives the possibility to choose another browser as default.

### Support of LOD actor is added (useful in SMESH and VISU module for large models)

SALOME VTK viewer now has a new functionality - management of Level Of Details (LOD) rendering through the usage of the native VTK LOD functionality (see also http://www.vtk.org/doc/release/4.2/html/classvtkLODActor.html). It allows the user to manage performance of rendering through the corresponding dialog. By default the dialog is disabled and the viewer behaves as before. To activate the functionality check "Enable" check box and click "Apply" or "Ok" buttons. In the dialog you can choose ether "Desired Update Rate" (by default 15 Frames Per Second) or "Still Update Rate" (by default 0.0001 FPS). The first parameter corresponds to the rendering when an action, for example: rotation, panning etc. is activated. The second parameter corresponds to the rate of restoring when an action is finished.

## Bug fixes and improvements

Total number of corrected external problems since SALOME 3.2.0a2: 17

The following external bugs have been corrected in the version:

- PAL5852 V1.4.1 : can't load PyQt component GUI
- PAL6839 Abort on GEOM with scale transform on first edge from a cone
- PAL10035 No warning on saving if pemissions are not enough
- PAL11554 Portability gcc4.0.1
- PAL11696 a MESH with FAMILY attributes (identifier, value, description).
- PAL11851 SIGSEGV in SALOME_PYQT_Module.cxx
- PAL12003 Problem when importing IGS geometry
- PAL12113 Can't visualize an edge in shading mode in OCC view
- PAL12114 createAndDisplayGO zoom and fit all
- PAL12117 How to implement Dump python in python modules
- PAL12120 SetShading not found in python
- PAL12149 Cannot create non-planar shells
- PAL12199 T2_2_9pre and previous versions: launch with standalone servers fails
- PAL12227 Dump script does not work
- PAL12236 SIGSEGV during init function definition of LoopNode.
- PAL12311 ExportToMED into a python script file
- PAL12004 VTK window => SIGFPE Arithmetic Exception Detected

The following improvements have been realized:

- PAL8238 Hypothesis GUI for non regular 1D meshing
- PAL9633 Save the study-view parameters in the hdf file
- PAL10462 Reactivate GEOM_CLIPPING plane functionality
- PAL10721 Saving the kind of representation in Post-pro preferences
- PAL11186 Management of the dockable GUI elements

In addition a lot of other bugs have been fixed during the GUI test campaign.

# Supported Linux distributions and pre-requisites

SALOME 3.2.0 beta1 supports the same Linux versions as SALOME 3.2.0a2 plus Mandriva 64 bit. Please note that SALOME is not certified in this OS.

*Note: OPEN CASCADE TECHNOLOGY version has been changed to OCCT 6.1 official version (can be downloaded from www.opencascade.com site).*

*New NETGEN 4.5 version comes with installation procedure and new latest doxygen version 1.4.6 which support generation of documentation from python.*

*HappyDoc is removed from third party products*

The SALOME 3.2.0beta1 version has been mainly tested with the following pre-requisite list on Mandrake 10.1 platform. However, other versions of pre-requisites may also work. The version was also tested on Debian 3.1 distribution and on native Mandrake10.1 prerequisites.

- gcc 3.4.1 - native for Mandrake 10.1
- tcl/tk 8.4.5 - native for Mandrake 10.1
- Python 2.3.4 - native for Mandrake 10.1
- Qt 3.3.3 - native for Mandrake 10.1
- Doxygen 1.4.6
- Boost 1.31.0 - native
- sip 4.1
- PyQt 3.13 *officially does not support Qt 3.3.3 but nevertheless it can be used*
- Swig 1.3.24
- OpenCASCADE TECHNOLOGY 6.1
- Qwt 4.2.0/0.2.4
- OmniORB 4.0.5
- OmniORBPy 2.5
- OmniNotify 2.1
- Hdf5 1.6.3
- Med 2.2.3
- Vtk 4.2.6
- Numeric 23.7
- Graphviz 2.2.1
- NETGEN 4.5 *a patched version of NETGEN*

*Note that the list of pre-requisites for Redhat 8 platforms is different and can be found within the Redhat 8 installation procedure for SALOME 3.2.0 b1that can be found on the FTP site.*

# How to get the version and pre-requisites

The SALOME 3.2.0b1 pre-compiled binaries for Mandrake 10.1, Debian Sarge, Mandriva 2006 and RedHat 8.0 can be retrieved from the PAL/SALOME FTP site (ftp://www.opencascade.com). Alternatively, SALOME modules can be downloaded from the following CVS repositories:

- KERNEL module: :pserver:<username>@cvs.opencascade.com:/home/server/cvs/KERNEL
- GUI module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/GUI
- GEOM module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/GEOM
- MESH module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/SMESH
- SUPERVISOR module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/SUPERV
- VISU module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/VISU
- MED module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/MED
- NETGEN plugin module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/NETGENPLUGIN
- SAMPLES: pserver:<username>@cvs.opencascade.com:/home/server/cvs/EXAMPLES

IMPORTANT! Source files for version 3.2.0beta1 are available in CVS via tag **V3_2_0b1**

The patch on NETGEN is placed inside NETGENPLUGIN sources. During the compilation of a plug-in, the patch is applied automatically to the standard NETGEN installation.

All other pre-requisites shall be obtained either from your Linux distribution (*please be sure to use a compatible version*) or from the distributors of these pre-requisites (*www.trolltech.com for QT for example*).

## Known problems and limitations

- Due to modifications in MED module some old MED files (for example, “fra.med”) can not be loaded any more (but they can be loaded in VISU). They have incorrect definition of Gauss points, as now MED has stricter rules of definition of MED data (bug PAL11767, 10770, 11947, 11580).
- Due to a bug with changes of tolerance in OCT 6.1, GUI test scenario PROD 04 can not be played in TUI mode without workaround. Workaround is to call the same Boolean operation cut twice (PAL8778)
- Partition algorithm has some problems on Debian Sarge. Due to this “Check geom” fails. (PAL11259, PAL12200)
- OCT 6.1 has problems in debug mode on Debian Sarge
- VISU module does not support timestamps defined on the same field but on different meshes
- In the current implementation of “Save VISU” state operation the parameters of Gauss view Partition mode are not stored. If a window has been partitioned and saved, it will be restored as non-partitioned. The same concerns background color .
- The second run of ex00_all.py on Debian leads to crash due to the above mentioned problems in partition algorithm (PAL9863)
- Display of some presentations on quadratic elements in VISU fails (cannot create animation for IsoLines, CutPlanes etc.) in the VTK, because the currently used version of the VTK library (4.2.6) can not properly process quadratic mesh elements (only ScalarMap and DeformedShape can be created) from the MED file. Unfortunately it is impossible to replace or extract the VTK functionality from the library. This problem will be fixed automatically when the SALOME platform will be ported to VTK 5.0 version or higher). This concerns Gauss viewer on quadratic elements. It is also impossible to build gauss presentations on some files with quadratic elements.
- PAL8065 still exists in 3.2.0b1 version (TestVisu20.xml fails in Supervisor)
- Step-by-step execution in SUPERVISOR fails on some graphs. This is only a prototype functionality which is not finished completely (PAL9906)
- Due to VTK 4.4 limitation, there are problems with the display of numbers of nodes in SMESH module (some numbers disappear from the viewer) PAL10412
- VTK presentations in GEOM have not been completely finished and cause problems with performance and memory usage. It desirable to use OCC viewer in GEOM module.
- MEFISTO algorithm fails on some cases (PAL11897, PAL12201)
- Due to some opened bugs, some non regression TUI tests fail:
    - MED component PAL11580 (004/A1), PAL11626 (002/A3) PAL10355 (003/A1, 004/A0 005/A1-I9, J1-J3) PAL10348 (006/A4, A5)
    - SMESH module: PAL11897, PAL12201 (MEFISTO fails) 001/A5, 002/C5
    - GEOM module, partition problems: PAL11259(011/A3), PAL12200 (003/F9)
- End user documentation for Supervisor module has been updated only with screenshots.
- Results of some TUI non-regression testing are different on slow and fast computers. This is due to using in some test cases the functionality of GEOM from Supervisor in parallel nodes. Due to the fact that Open CASCADE Technology does not thread safe, in some hardware configurations there is parallel conflict access to some data and such supervisor graphs fail. This problem will be fixed in future. At this moment the workaround is usage of GEOM nodes subsequently.

## Maintenance caution!

No maintenance is provided for this version and this is not the version that will be used by end users. The functionalities of SALOME 3.2.0 beta1, its content and API may change and will be developed into the SALOME 3.2.0 as a maintainable version. Please contact OPEN CASCADE PAL/SALOME team for any questions.

Copyright © 2001-2006 All rights reserved.