

## General information

OPEN CASCADE is pleased to announce SALOME 3.1.0 major development version. It is a development release that contains the results of planned improvements and bug fixes with reference to the SALOME 2.2.2 version released in the beginning of year 2005.

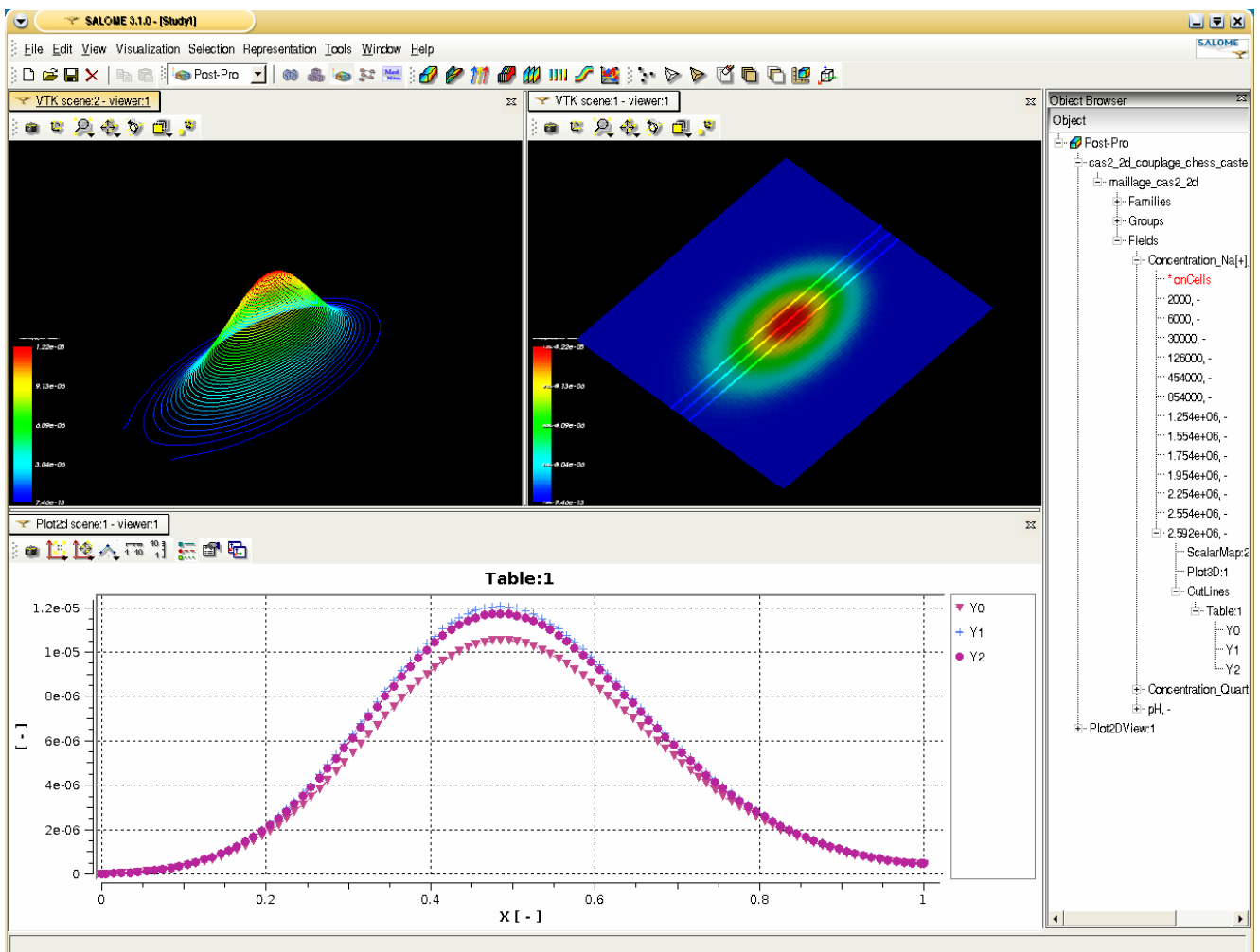
The major goal of SALOME 3.1.0 is to demonstrate the state of development carried out during year 2005 by CEA/EDF/OPEN CASCADE. This document describes the main differences since SALOME version 2.2.2 .

## New features (since SALOME 2.2.2)

### General modifications

#### *New look and feel*

SALOME 3.1.0 look-and-feel has been significantly improved. Please find below precise explanations concerning all changes and improvements.



#### *Specific GUI style*

Special own GUI style has been developed for SALOME 3.1.0. The style differs from standard KDE styles in the following aspects:

- New menu design; Gradient fill in menus and toolbars;
- Non-active buttons displayed in transparent grey scale;
- New and simple look for all controls;
- Buttons are highlighted when the mouse cursor points at them;
- Current dockable window has active header
- Gradient fill in the headers of QListView class;

- New look of splitter in class QSplitter;
- New look of tabs in class QTabWidget.

### *Multi-desktop approach*

A Multi-desktop GUI is first applied in SALOME 3.1.0. In fact, each document (Study) has a separate Desktop. Users feel more comfortable and will not be confused because now all windows connected to a certain Study (viewers, dockable windows etc) are placed in one container. This approach is much better than the classic multi-document interface in case if many windows are open, i.e. if one document contains a lot of viewers, dialogs, etc. Each desktop also has its own menu and toolbar.

### *Dockable toolbars and panels*

In SALOME 2.x.x the desktop space was distributed among windows through a special container – splitter. It allowed changing the sizes of the windows on the desktop. In SALOME 3.1.0 almost all windows have been redesigned to use the dockable mechanism. Dockable windows and toolbars can be placed in special areas, on the borders of the desktop. The size and position of dockable windows can be changed by the user. He can even place such windows outside the desktop or hide them. Dockable windows are Object Browser, Python Console, MenuBar, etc.

### *Tabbed viewers*

There is a new window manager in SALOME 3.1.0. As opposed to manager from 2.x.x version, the new one displays only one active window in the representation area. All windows are placed inside tabs where the user can switch from one to another. If there are several windows and the user wants to display more than one at the same time, he can split the representation area in two parts, in the horizontal or vertical direction; this action creates two representation areas, instead of one, with windows placed in them. The size of such representation areas can be changed by the user. He can also move any windows between representation areas. When an area becomes empty, it is not displayed.

### *New user preferences GUI (menu File / Preferences)*

A new mechanism for working with preferences and resources is introduced in SALOME 3.1.0. Now the editor of preferences is common for all modules and it uses only one dialog box where users can customize preferences for each module. When the user changes the setting (clicks the “OK” button) each module is notified about the preferences that has been changed. The users can also edit preferences in module-specific windows.

### *Automated saving of look and feel preferences*

In 3.1.0 version all positions and sizes of dockable windows are stored in a special user file. Each time the user quits the application, the window coordinates are automatically saved in this file to be restored at the next start of the application. Also there are differences in the settings of the same windows (for example, Python console) between different modules. It means that the same window can have different size and position when the other module is activated. Each module can also indicate which common windows are necessary for working, so that only they could be displayed when this module is active.

### *Light components (engine-independent)*

Thanks to the architectural changes in the SALOME GUI, SALOME 3.1.0 version enables to create engine-independent components. These components may not use Corba at all, having the internal data structure, which can be written in pure C++ (or python). Such modules are located inside the SALOME GUI process and from the end user point of view do not differ from standard components. The “LIGHT” component, an example of such components, is placed into EXAMPLES CVS.

### *Dump study (dump Python)*

A special feature to create a python script which can rebuild all data from a SALOME study has been implemented. Now a study created by the GUI can be written into a python script for automation. This feature is supported in GEOM, SMESH, VISU and SUPERV modules and can be extended to any module based on the SALOME engine.

### *Copyright updates*

SALOME copyright information was updated and normalized in all files where it had been previously absent. This action is done according to the PAL/SALOME Technical Committee 11 decision and according to SALOME 2 partner agreements.

## ***Dependency on MED library removed from KERNEL***

MED Wrapper library has been moved from KERNEL to MED module. Thus, SALOME KERNEL no longer depends on the MED library. This change has also affected other SALOME modules. In particular, configuration procedures have been updated for standard SALOME modules, which use the MED library. They detect the MED library presence themselves, rather than relying on KERNEL means as it was done previously.

## ***Examples data files movement***

Examples data files from MODULE\_ROOT\_DIR/examples were regrouped in a separate folder according to file types. Corresponding CVS module SAMPLES\_SRC is created under the CVSROOT EXAMPLES. Its structure is the following:

- SAMPLES\_SRC – root directory
- SAMPLES\_SRC/MedFiles – subdirectory for MED files
- SAMPLES\_SRC/Unv – subdirectory for unv files
- SAMPLES\_SRC/Sauv – subdirectory for sauv files
- SAMPLES\_SRC/inp\_xyz – subdirectory for inp and xyz from MED module
- SAMPLES\_SRC/Shapes – subdirectory for geom shapes (in BREP, IGES and STEP formats)
- SAMPLES\_SRC/Shapes/Brep – subdirectory for BREP files
- SAMPLES\_SRC/Shapes/Iges – subdirectory for IGES files
- SAMPLES\_SRC/Shapes/Step – subdirectory for STEP files
- SAMPLES\_SRC/Tables – subdirectory for VISU table files
- SAMPLES\_SRC/Superv/ – subdirectory for supervisor example files
- SAMPLES\_SRC/Superv/Python – subdirectory for python scripts
- SAMPLES\_SRC/Superv/Graphs – subdirectory for XML files which contains graphs for Supervision

The corresponding files were removed from other SALOME modules except for the MED component. Files in the MED\_ROOT\_DIR/resources were left in order to support current test mechanism for the MED component.

A new environment variable DATA\_DIR has been introduced to indicate the location of SAMPLES\_SRC. All corresponded python file examples were updated to support new placement of data files. All TUI tests were updated too.

## **GUI module**

### ***Customization of viewers for modules***

Existing viewers (Open Cascade viewer 3d, VTK viewer 3d, Qwt plot , etc) have been separated into different libraries. All of them are written using the base interface and have similar structure. Each module can ask the application to create a copy of a predefined viewer or can add its own viewer inherited from the base interface into the application.

### ***GUI of modules as plug-ins***

All GUI of standard modules in SALOME 3.0.0 have a plug-in structure, i.e. each module is placed in the dynamic library which is loaded on demand. All modules can create their own menu items, buttons in toolbars, windows etc.

### ***Ability to build non-SALOME applications (CAM level)***

There is a special separate CAM level, used for writing modules which don't use standard tools of the SALOME platform. The CAM component serves as the basis for the new SALOME GUI and contains all basic functionality for working with modules (loading a module; opening, saving, closing a document, customization of toolbars and menus).

### ***Other improvements***

- Handling of references has been changed: by default, in the current version the user should remove invalid references to a deleted referenced object manually (through a popup). However each module can redefine such behavior and implement automatic deletion of references and removal of referenced objects.
- New improvement PAL10635 is realized a in a tabbed viewer. In the 3.1.0beta1 version it is possible to rename and close viewers through popup menus.
- Display and Erase in a popup present only for objects which can be displayed in a current active viewer.

- Improvements in behavior of SALOME preferences
- New improvements have been made in the object browser. Now double click on a reference object changes the selection to the original object. Also there is a possibility to work with references like with separate objects and introduce different logic comparing to the original object.

## **KERNEL improvements**

### *Separation and repackaging*

The KERNEL module has been repackaged for 3.1.0 version and from now KERNEL and the GUI are separated. KERNEL provides basic functionalities for Container and Component management, SALOME Data Structure implementation, Life Cycle mechanism etc. The GUI module deals with definition of interactive services, i.e. Qt written application (desktop, basic dialogs etc), viewers (VTK, OpenCascade, etc), selection mechanism etc. It allows writing pure batch components which use only the KERNEL module, for example, for batch mode parallel computations.

### *SALOME DS proxy*

A concept of SALOMEDS proxy has been implemented in Salome version 3.0.0. in order to optimize the usage of SALOMEDS services and facilitate the development of GUI components without CORBA. The modifications group in two main parts:

- Repackaging of the SALOMEDS package. The SALOMEDS package has been divided in two 2 packages: SALOMEDSImpl - this package contains a pure C++ (no CORBA involved) implementation of SALOMEDS services and SALOMEDS - the package containing CORBA wrappers to the services implemented in the package SALOMEDSImpl
- Creation of special wrappers, which can be used for calling SALOMEDS services. The wrappers are declared as abstract C++ interface classes, whose structures and declarations coincide with the interfaces given in the IDL description of SALOMEDS.

These abstract interfaces are presently implemented for SALOMEDS objects. All calls to SALOMEDS are hidden in the interface.

Hiding the calls to the real implementation allows the developer to increase the performance while using the SALOMEDS services in case when the SALOMEDS server and its clients are running in the same address space. In this case all calls to the SALOMEDS services are directed to the local implantation in SALOMEDSImpl without involving the CORBA mechanism.

### *Other KERNEL improvements*

- The SALOME 3.1.0 includes major modifications done by EDF including a redesign of message traces and introduction of unitary tests based on the cppunit utility.
- Version 3.1.0 includes KERNEL modifications related to LifeCycle CORBA and SALOME Application concept done by EDF.
- Various KERNEL improvements done by EDF are within this version

## **GEOM module**

### *New GEOM plugin for import of CATIA V5 files*

OPEN CASCADE is glad to present a new plugin for import of native CATIAV5 files into the GEOM module. To retrieve the sources of the plugin you can take them from the CVS server:<username>@212.37.193.42:/home/server/cvs/CATIAV5PLUGIN.

*Important remark: The plugin is a wrapper for a library which converts CATIA files – this library is a commercial product protected by license. To receive an evaluation version of the converter please contact OPEN CASCADE SAS.*

### *Other GEOM improvements*

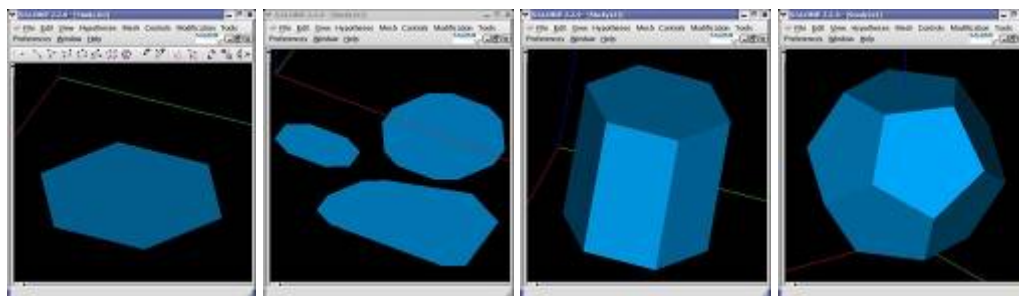
- Repackaging of GEOM modules packages has been performed in order to produce non-CORBA packages to simplify the engine architecture of this module
- Implementation of request PAL10015. Like the « GetShapesOnXXX » functions specialized to build geometric group, GetShapesOnQuadrangle is a new function that helps to build 2D geometric groups.

## MESH module

### *Polygon and polyhedral support*

The SMESH module has been updated to add the possibility for creation of polygons and polyhedral elements

Operations on mesh now support the creation of polyhedral elements (sewing, pattern mapping, extrusion, revolution etc)



### *General redesign of GUI for mesh creation*

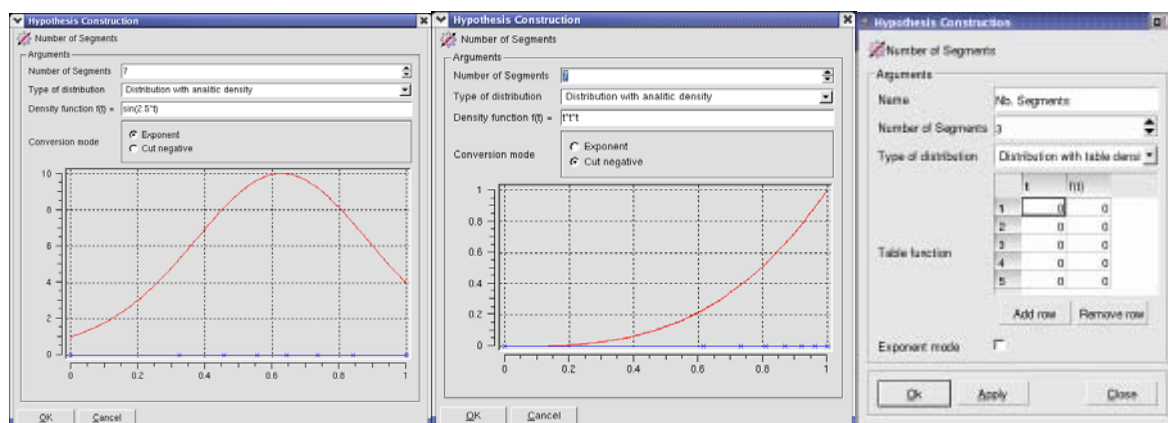
General redesign of GUI for mesh creation in this module. This was done to simplify the procedure of creation of hypotheses. Now all operations are available in one dialog and the user can switch through tab panels to choose the necessary algorithms and hypotheses (see the picture)



### *Improvement of 1D hypothesis "Number of segments"*

The 1D hypothesis "Number of segments" is now more generalized, providing a powerful functionality for creation of points on edges using the following distributions:

- Regular distribution of points
- Non regular using scale factor (arithmetic progression)
- Table distribution
- Analytical formula for point distribution
- Preview for tables and analytical presentation - draw of function and distribution of points
- Numerical algorithm for points calculation based on density function is improved to provide robust results





## *Other minor improvements*

- Functionality for mesh extrusion was improved and currently includes parameters for automatic Sewing calling after the extrusion of 2D mesh and option to switch on/off the building of 2D skin mesh.
- After the implementation of improvement PAL8377, export to the med dialog box now allows controlling the creation of default groups of elements in med files
- Portability with gfortran of MEFISTO algorithm (patch from bug PAL10718) has been applied
- In addition the functionality of SMESH module accessible from GUI was revised to be coherent with all available functions of the TUI mode. As a result, now the GUI of TriangleToQuadrangle and QuadrangleToTriangles operations includes additional options giving a choice from the two possible variants, depending on the quality (for example, aspect ratio) and the maximum angle allowed between two triangles to unite them in a quadrangle.
- SMESH now provides computation of volume mesh by NETGEN without hypothesis by default
- Implementation of request PAL10199. Introduction of new SMESH 3D Control: volume of the tetra/hexa.
- New feature is available - automatic 1D mesh creation. Automatic length hypothesis provides a possibility to generate default mesh based on minimal length of the edge and bounding box of model.

## **MED module**

### *Polyhedral and polygon elements support.*

MED wrapper class in KERNEL has been updated to support reading and writing of polyhedral elements. IDL interfaces as well as MEDMEM have been modified in order to support this new feature.

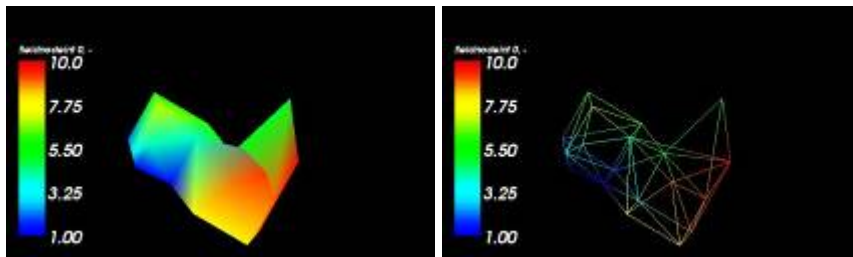
## *Other improvements*

- Compilation of MED component without connection to KERNEL. It's done to have a standalone MED component (mainly MEDMEM) for other projects. It can be done using options '--without-kernel' or '--with-kernel=no' for build\_configure and configure steps.
  - From now on the MED module can save its data within the SALOME study. This allows external modules to write down their MEDMEM fields published in the study
- Different improvements done by CEA are included in version 3.1.0

## **VISU module**

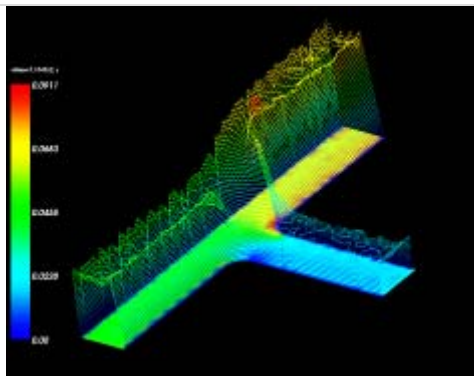
### *Polygon and polyhedral support*

- VISU module can visualize polyhedral elements and data field on them
- VTK visualization classes have been updated to support the display of polygons/polyhedrons

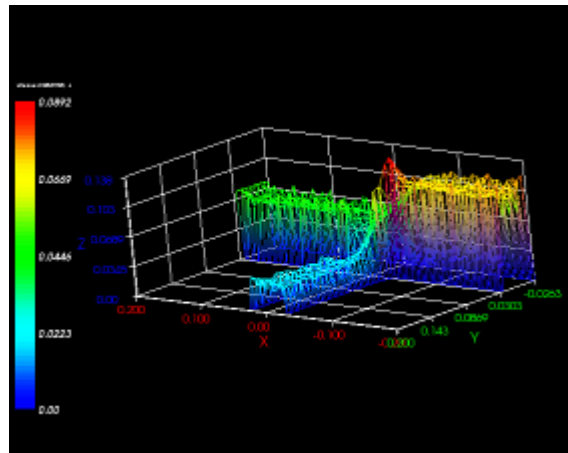


## *Other major improvements in the VISU module*

- Clipping plane. Like the SMESH module, VISU now has “clipping plane presentation”. It includes classical clipping and also i-j-k in case of a structured mesh.
- Merging scalar bars. VISU allows to have global scalar bar for different presentations now
- 2D fields displayed in 3D. New kind of presentation has been introduced – it builds 3D presentation based on assigned fields for mesh



- Translation of presentations. The user can translate presentations in 3D viewer in order to see different results simultaneously in one space
- New animation mode based on translated presentations
- Graduated bounding boxes. Graduated rules are accessible through a menu now – it allows to see the dimension of presentations



### *Other minor VISU improvements*

- Multiple delete operation
- Implementation of request PAL8856, now set axis scale is available from TUI
- Implementation of request PAL9160, currently shading for visualization in VTK viewer can be controlled through preferences
- In the framework of PAL10401 bugfix a new improvement was introduced for creation on cut lines and curves based on them
- There are new features: preview of order of cut lines using the marker in the viewer and also a possibility to reverse them.

## **SUPERVISOR module**

Implementation of improvements request PAL8551 for Supervision messages classification was done

## **Installation procedure improvements**

- Open Cascade Technology prerequisite have been changed to 6.0 pre-version
- Examples (SAMPLES\_SRC) have been added
- Compilation of SALOME from sources function is added. Now it's possible to choose compilation of SALOME from sources
- Compilation of Open Cascade Technology from sources is added
- Bug with generated libtool \*.la files has been corrected. Now paths in \*.la files are modified during the installation in order to achieve usage of libtool during the compilation of 3<sup>rd</sup>-party products

## **General documentation improvements**

### *GUI source files documented*

Special efforts were made in part of documenting GUI source files. Comments were added for most important classes and methods. With version 3.1.0, it should become easier for developers to understand the principles of GUI framework introduced in version 3.1.0.

Renovated and significantly improved documentation of a part of modules was added providing examples and TUI command interface description.

In general please have a look at the documentation of installation of/running SALOME, its general desktop organization, KERNEL operations and end user actions for SMESH and GEOM modules with the corresponding examples of TUI scripts.

These help items are available via SALOME Help menu and provide access to Python (TUI) documentation as well.

### *New samples*

New sample LIGHT has been developed by OCC. It is called [LIGHT\\_SRC](#) and placed in EXAMPLES CVS. This module demonstrates how to write a simple corba-light component.

New sample developed by CEA has been adopted for SALOME 3.0.2. It is called [CALCULATOR\\_SRC](#) and it demonstrates how to write a C++ Salome Module which is interfaced with MED Memory, using MedClients classes. Associated documentation in swx and pdf format can be found in the /doc directory.

New sample SIERPINSKY developed by OCC is adopted for SALOME 3.1.0a2. It is called [SIERPINSKY\\_SRC](#) and [RANDOMIZER\\_SRC](#) and it demonstrates how to write a simple C++ and Python Salome Module which is interfaced with both VTK viewer and MED component. The example implements a simple interface to calculate Sierpinsky fields. Please read README in SIERPINSKY\_SRC about the configuration and usage of this example  
It can be retrieved from CVS EXAMPLES :pserver:<username>@cvs.opencascade.com:/home/server/cvs/EXAMPLES.  
Module name is [SIERPINSKY\\_SRC](#) and [RANDOMIZER\\_SRC](#).

## Supported Linux distributions and pre-requisites

[SALOME 3.1.0](#) supports the Mandrake 10.1, Debian Sarge, Feodora 3 and RedHat 8.0 Linux versions  
The [SALOME 3.1.0](#) 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 1.3.7 - native for Mandrake 10.1
- Boost 1.31.0
- 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 6.0
- Qwt 4.2.0/0.2.4
- OmniORB 4.0.5
- OmniORBPY 2.5
- OmniNotify 2.1
- Hdf5 1.6.3
- Med 2.2.2
- Vtk 4.2.6
- Numeric 23.7
- HappyDoc 2.1
- Graphviz 2.2.1
- NETGEN 4.3 a patched version of NETGEN

*Note that the list of pre-requisites for Redhat 8 and Debian 3.1 platforms are different and can be found within the Redhat 8 installation procedure for SALOME 3.1.0.*

## How to get the version and pre-requisites

The [SALOME 3.1.0](#) pre-compiled binaries for Mandrake 10.1 and RedHat 8.0 can be retrieved from the PAL/SALOME FTP site (<ftp://ftp.opencascade.com>). Alternatively, SALOME modules can be downloaded from the following CVS repositories:

**KERNEL** module: :pserver:<username>@cvs.opencascade.com:/home/server/cvs/KERNEL  
The name of module KERNEL\_SRC

**GUI** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/GUI  
The name of module GUI\_SRC



**GEOM** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/GEOM

The name of module GEOM\_SRC

**MESH** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/SMESH

The name of module SMESH\_SRC

**SUPERVISOR** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/SUPERV

The name of module SUPERVISOR\_SRC

**VISU** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/VISU

The name of module VISU\_SRC

**MED** module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/MED

The name of module MED\_SRC

**NETGEN** plugin module: pserver:<username>@cvs.opencascade.com:/home/server/cvs/NETGENPLUGIN

The name of module NETGENPLUGIN\_SRC

**SAMPLES – salome data files** : pserver:<username>@cvs.opencascade.com:/home/server/cvs/EXAMPLES

The name of modules SAMPLES\_SRC

**EXAMPLE modules** : pserver:<username>@cvs.opencascade.com:/home/server/cvs/EXAMPLES

**SIERPINSKY example** – SIERPINSKY\_SRC and RANDOMIZER\_SRC

**HELLO example** – HELLO1\_SRC

**PYHELLO example** – PYHELLO1\_SRC

**COMPONENT examples** – COMPONENT\_SRC

**LIGHT example** – LIGHT\_SRC

**CALCULATOR examples** – CALCULATOR\_SRC

**GHS3DPLUGIN** : pserver:<username>@cvs.opencascade.com:/home/server/cvs-PAL/GHS3DPLUGIN

The name of module – GHS3DPLUGIN\_SRC

**IMPORTANT!** Source files for version 3.1.0 are available in CVS via tag **V3\_1\_0**

**Open Cascade Technology 6.0** can be retrieved in its binary and source forms from the FTP site (<ftp://www.opencascade.com>) or can be found on a CD with SALOME 3.1.0.

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**

- Please, note that this version is a development release. It has not gone through complete testing and debugging processes.
- There is a well known problem with NVidia drivers on some configurations. It leads to SIGFPE exceptions during the display of scalar map presentations in the VTK viewer. To fix this problem it is necessary to install latest drivers for video cards
- Due to some opened bugs, some non regression TUI tests fail:
  - MED tests: PAL10355 (003A1, 004A0), PAL8701 (005A1-I9, J1-J3), PAL10348(006A4,A5)
  - COMPONENT tests: PAL10730(001 C1,C2), PAL10733(001 C3, C7)
- End user documentation for Supervisor and VISU modules were updated only in parts of screenshots. No revision of these modules was done due to planned major changes in GUI for SALOME 3.2.0 version
- Results of some SUPERVISOR TUI non-regression tests 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, on some hardware configurations there is parallel conflict access to some data and such supervisor graphs fail. This problem will be fixed in the future. At this moment the workaround is using GEOM nodes subsequently.
- Advanced features of 1D hypothesis “Number of segments” for SMESH have not been completely finalized yet. However it provides quite a good level of functionality to start its testing.

## **! Maintenance caution!**

This is a development release of SALOME thus no maintenance is provided for this version and this is not the version that shall be used by end users. The functionalities of SALOME 3.1.0, its content and API may change and will evolve into the SALOME 3.2.0 as a maintainable version (provisioned for the end of April 2006) .Please contact OPEN CASCADE PAL/SALOME team if you have any questions.

