

SALOME 5.1.4

Public maintenance release announcement

June 2010



GENERAL INFORMATION

CEA/DEN, EDF R&D and OPEN CASCADE are pleased to announce [SALOME](#) version [5.1.4](#). It is a public maintenance release that contains the results of planned major and minor improvements and bug fixes against SALOME version 5.1.3 released in December 2009.

Table of Contents

- **GENERAL INFORMATION1**
- **NEW FEATURES AND IMPROVEMENTS3**
 - PREREQUISITES CHANGES (COMPARING WITH SALOME 5.1.3).....3
 - LICENSE RESTRICTIONS3
 - NEW MODULES3
 - IMPROVEMENTS4
 - New Geometry object: T-shape Pipe*.....4
 - Improved Filling Surface with Edges operation*4
 - Compound of Points by Intersection of 2 Wires*.....5
 - Find Element by Point*.....5
 - Right and Left axes for Curves from tables*.....6
 - Sort Table Data*.....7
 - Split into tetrahedrons*.....7
 - Point Marker for Mesh and Post-Pro modules*8
 - Mesh Scaling*.....9
 - Mesh computation priority*.....10
 - Precision*11
 - YACS improvements*.....11
 - Other Improvements*.....12
- **BUG CORRECTIONS15**
 - GUI MODULE (IAPP)15
 - KERNEL MODULE15
 - GEOM MODULE15
 - SMESH MODULE.....18
 - VISU MODULE.....22
 - MED MODULE23
 - YACS MODULE25
 - NETGENPLUGIN MODULE.....26
 - HEXOTICPLUGIN MODULE26
 - GHS3DPLUGIN MODULE.....26
 - BLSURFPLUGIN MODULE.....26
 - YACSGEN TOOL27
 - INSTALLATION PROCEDURE27
 - BUILD PROCEDURE.....27
- **CASCADE 6.3 SERVICE PACK 9 BUG CORRECTIONS28**
- **SUPPORTED LINUX DISTRIBUTIONS AND PRE-REQUISITES30**
- **HOW TO INSTALL AND BUILD SALOME32**
- **SALOME SYSTEM REQUIREMENTS32**
- **HOW TO GET THE VERSION AND PRE-REQUISITES.....32**
- **KNOWN PROBLEMS AND LIMITATIONS33**



NEW FEATURES AND IMPROVEMENTS

PREREQUISITES CHANGES (COMPARING WITH SALOME 5.1.3)

Open CASCADE Technology 6.3.0 service pack 8 has been replaced by 6.3.0 service pack 9 that includes a correction of the bugs that came from SALOME platform. See Change Log file in the source archive of the CASCADE 6.3.0sp9 for more details.

NOTE: it is necessary to recompile NETGEN mesher with the new Open CASCADE Technology version.

Additional changes of the 3rd-party pre-requisite products are listed below.

Removed pre-requisites

- **Numeric 24.2** has been removed from the list of prerequisites.

New pre-requisites

- **numpy 1.3.0** has been added to the SALOME pre-requisites list. It has replaced obsolete **Numeric** package. The **numpy** is a fundamental package for scientific computing with Python.
- **lapack 3.2** (3.1.1 for Debian 3.1 Sarge) has been added to the list of pre-requisites, as it is required by the **numpy**.

Updated pre-requisites

- VTK 5.0.0 has been replaced by **VTK 5.0.4**
- OCCT 6.3.0 sp8 has been replaced by **OCCT 6.3.0 sp9** (see above)
- libBatch 1.0.0 has been replaced by **libBatch 1.1.0**

For full list of pre-requisite products please refer to the chapter **Supported Linux distributions and pre-requisites** below.

LICENSE RESTRICTIONS

- Hereby we explicitly declare that PyQt 4 toolkit (Riverbank Computing Ltd) is distributed under the terms of GPL license.

NEW MODULES

SALOME 5.1.4 includes two additional modules:

- **JOBMANAGER** - this module is an interface to use SALOME Launcher services.
- **YACSGEN** - component and module generator for SALOME. It is a Python package (module_generator) to generate a SALOME module automatically from a short description of the components it shall contain. That description is done with the python language.

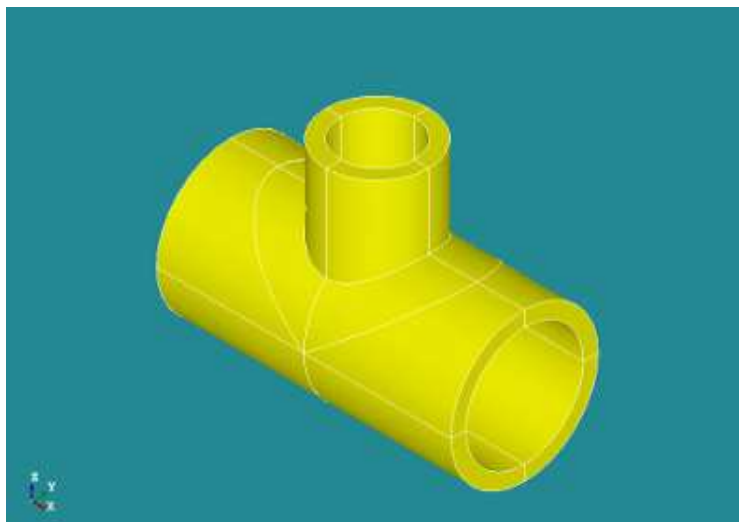
IMPROVEMENTS

New Geometry object: T-shape Pipe

A special constructor for a new **T-shape Pipe** object has been added to the Geometry module.

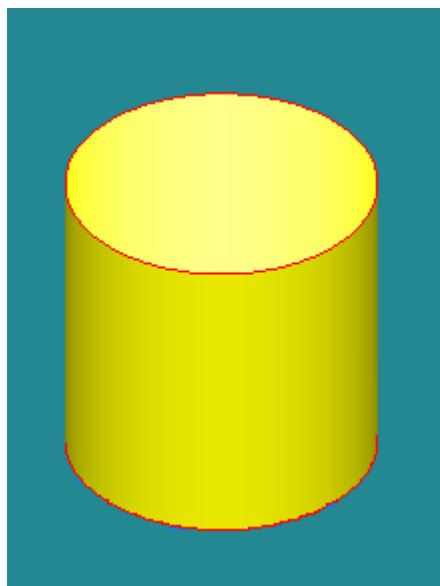
A **T-shape** pipe can be defined via the radius, length and thickness of the main and the incident pipes and three points defining the coordinates of the pipes junction.

It is also possible to add chamfer or fillet at the pipes junction.



T-shape pipe object

Improved Filling Surface with Edges operation



It is now possible to use closed wires as arguments for filling, thus obtaining closed pipes

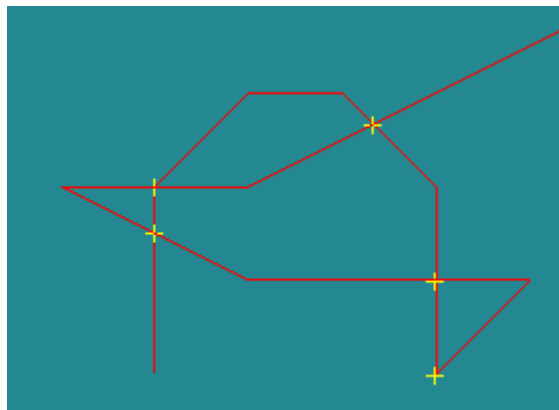
In the image, the yellow cylinder is the result of a filling of two wires shown in red.

Compound of Points by Intersection of 2 Wires

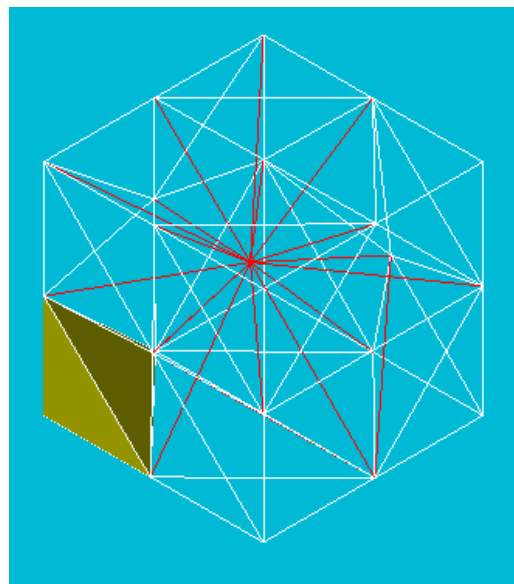
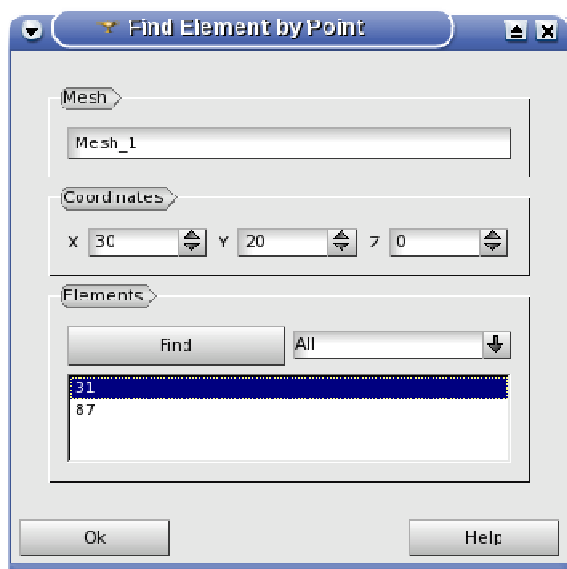
It is now possible to create a vertex defined at the 2 wires or wire and edge intersections.

If there are several intersection points, a compound of vertices will be created.

In the figure, the intersecting wires are red and the resulting points are yellow.



Find Element by Point

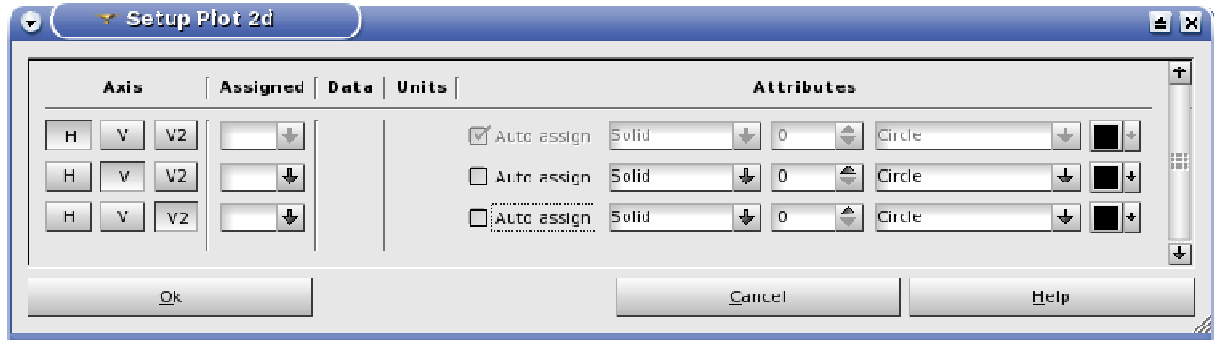


It is now possible to get the identifier of a 1D, 2D or 3D element by the coordinates of a point, that belongs to it. If several such elements are detected, all of them are listed in the corresponding list widget of the dialog box.

The element selected in the list of elements is highlighted in the 3D viewer.

Right and Left axes for Curves from tables

It is now possible to assign 2D curves presentations to any of two Y-axes: the right and the left when curves are created from the tables in the Post-Pro module.

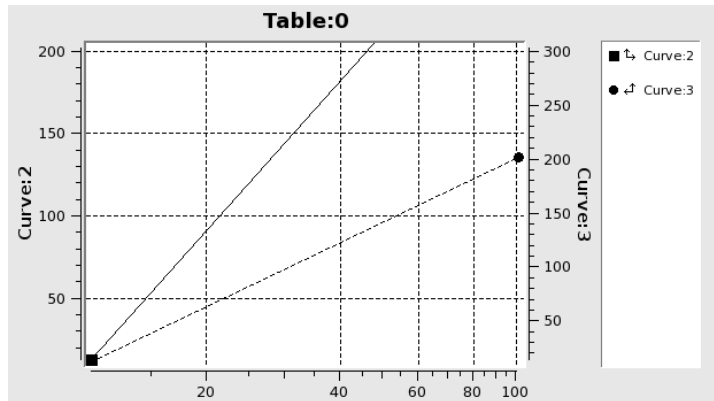


Setup Curves dialog has been extended with V2 button for choosing the right Y axis for the corresponding curve.

This option is also accessible via VISU engine IDL interface using `CreateCurveWithZExt()` method with an additional parameter.

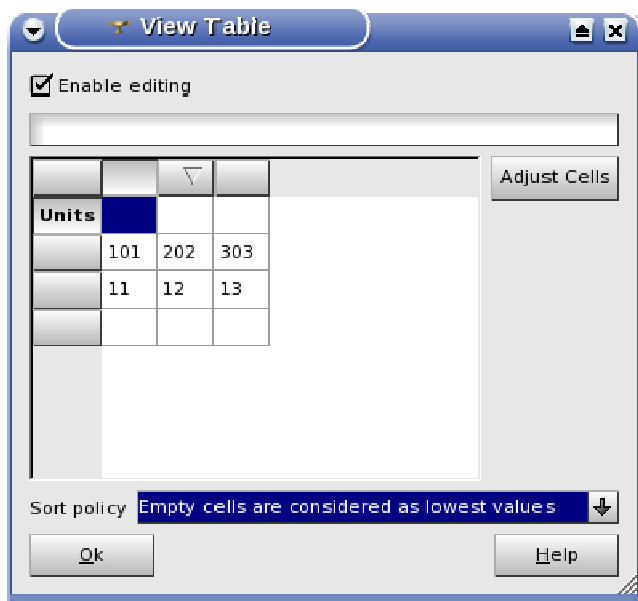
Visual behavior of the Plot2d view has been slightly changed due to this improvement.

Firstly, only the axes that are used by the visible curves are displayed. Thus, if there are no curves in the view, no axes are actually shown.



Secondly, when several curves displayed in the view are attached both to the left and the right vertical axes, the corresponding indicators are shown in the legend of the Plot2d view (see figure above).

Sort Table Data



It is now possible to sort the data in the tables. For this, switch on the *Enable editing* check box in the *View Table* dialog box and click on the header of the target column. First click sorts items in the ascending order; another click on the same column header reverses the sorting order.

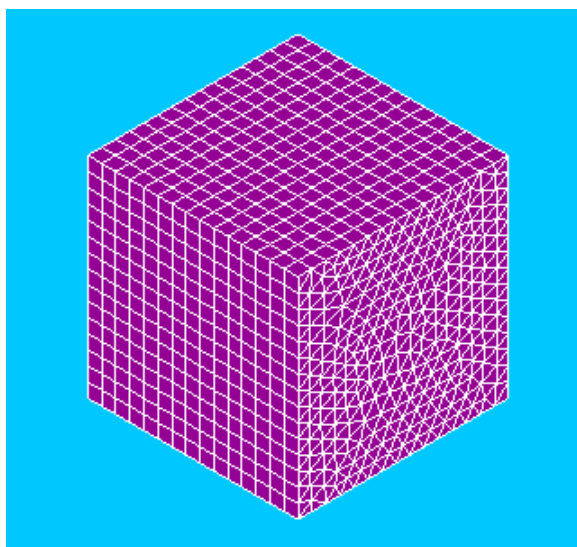
The *Sort policy* option specifies how the empty cells should be processed (if there are any in the table). The following modes are available:

- Empty cells are the lowest values
- Empty cells are the highest values
- Empty cells are always first
- Empty cells are always last
- Empty cells are ignored (means that the positions of empty cells will not change after sorting operation).

When the table is sorted, all curve presentations built using the table data are updated and redisplayed.

This functionality is also available in batch mode using `SortRow()`, `SortColumn()`, `SortByRow()` and `SortByColumn()` methods of the `VISU::Table` interface.

Split into tetrahedrons



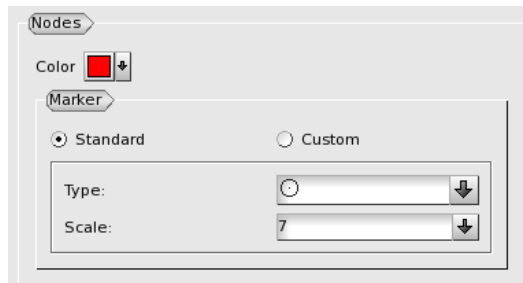
A new *Split into Tetrahedrons* modification operation is not available in the Mesh module. This operation splits all non-tetrahedral 3D mesh elements into tetrahedral ones, modifying 2D mesh accordingly.

In the figure there is a box with hexahedral meshing split into tetrahedrons on one of its faces.

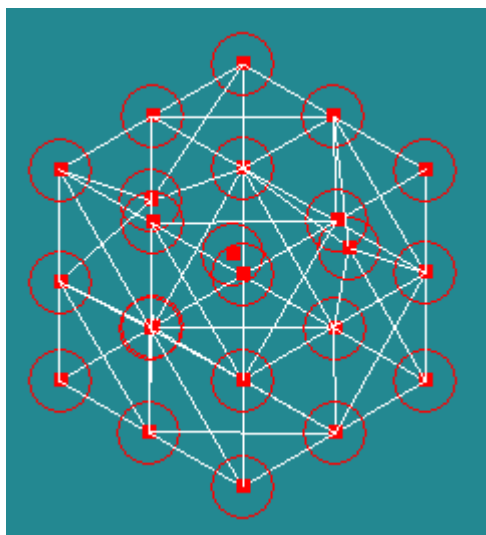
Point Marker for Mesh and Post-Pro modules

It is now possible to change the representation of the points in the 3D viewer either by selecting one of the predefined shapes or by the loading of a custom texture from an external file. This functionality is available in the Mesh and Post-Pro modules.

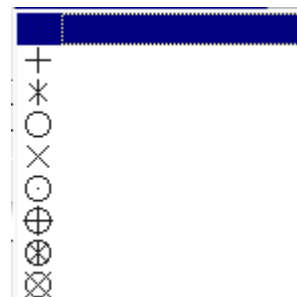
The external file should provide a description of the point texture as a set of lines; each line is represented as a sequence of "0" and "1" symbols, where "1" symbol means an opaque pixel and "0" symbol means a transparent pixel. The width of the texture corresponds to the length of the longest line in the file, expanded to the nearest byte-aligned value. The height of the texture is equal to the number of non-empty lines in the file. Note that missing symbols are replaced by "0".



See the mesh with one of the standard node markers in the figure below.



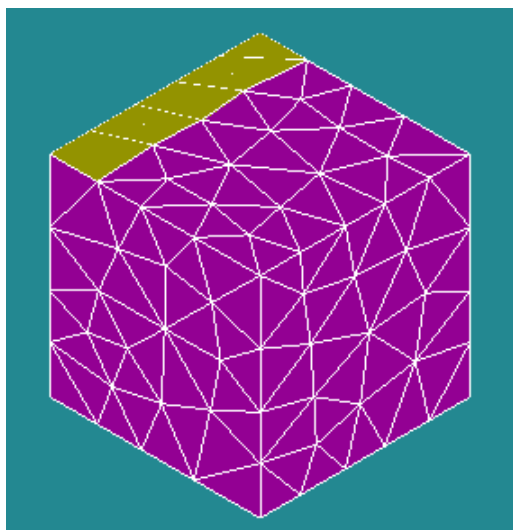
The Mesh and Post-Pro modules provide a set of predefined point marker shapes which can be used to display points in the 3D viewer. Each standard point marker has two attributes: type (defines shape form) and scale factor (defines shape size).



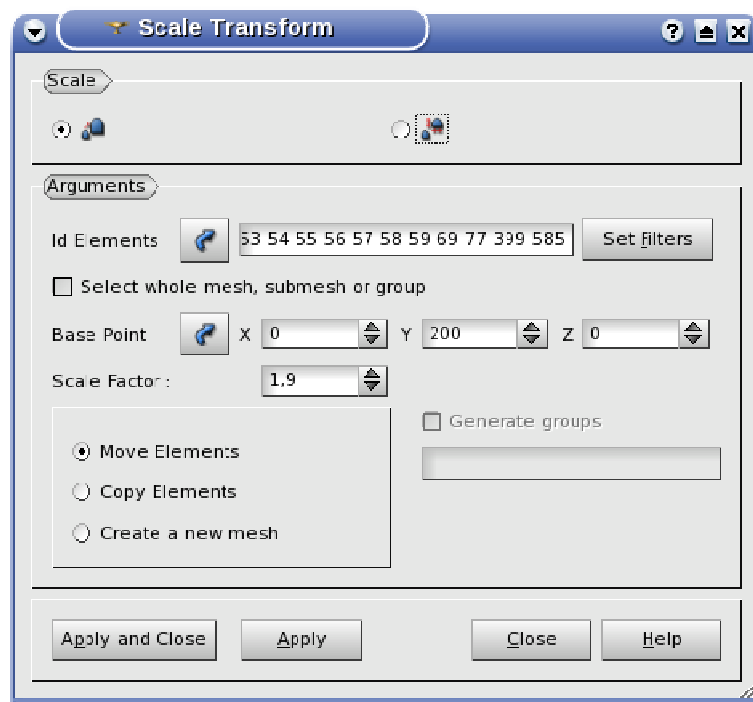
Mesh Scaling

This new **Mesh Scaling** functionality, available in the main menu via *Modification* → *Transformation* → *Scale transform*, allows scaling of the meshes or their elements

In the dialog box it is possible to select the whole mesh, submesh or group by activating the corresponding checkbox, to choose mesh elements with the mouse in the 3D Viewer or to apply a filter to the selection of elements.



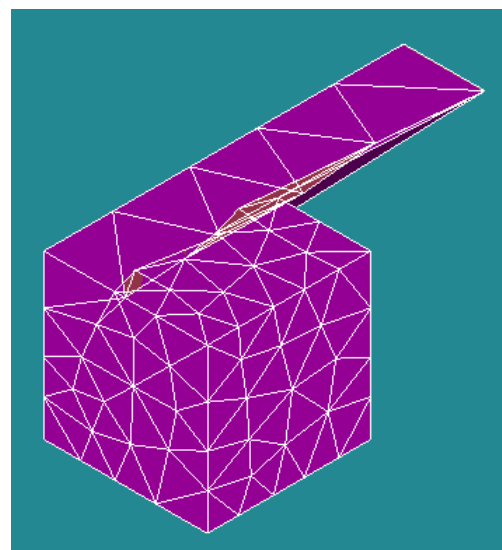
In the figure above there is an example of a box with highlighted elements selected for the scaling. It is possible to select a whole area with a mouse frame; or input the element IDs directly in ID Elements field.



In the dialog box, it is necessary to choose the base point relatively to which the elements are scaled and the scale factor.

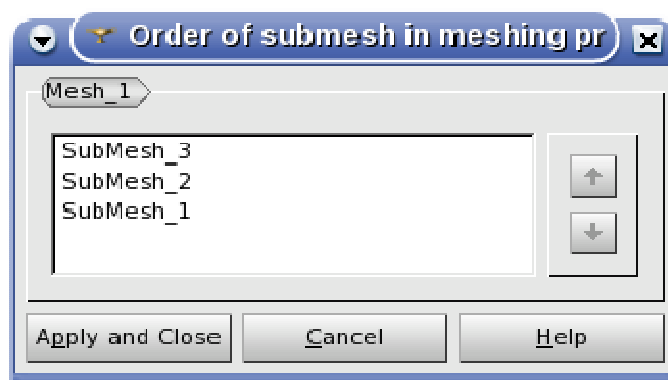
The scaled elements or mesh can be Moved (erased from the previous location) or Copied (left at the previous location) - the source mesh will be considered one and single mesh with the result of the scale operation. If "Create as new mesh" option is chosen, the scaling produces a new mesh with the default name MeshName_scaled.

To the right there is an example of the resulting mesh, created with *Move Elements* option.



Mesh computation priority

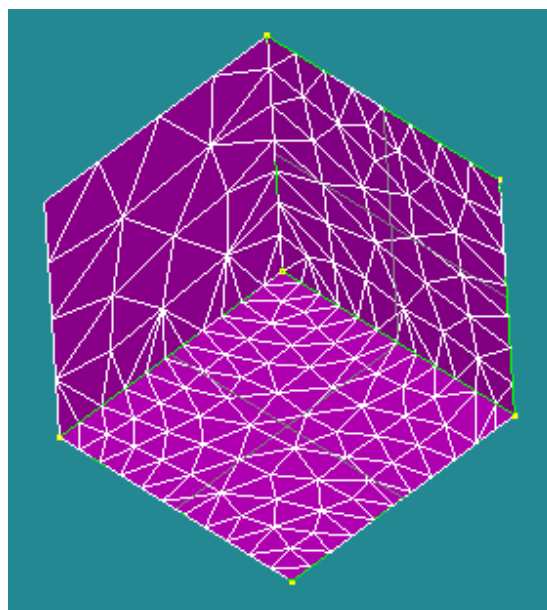
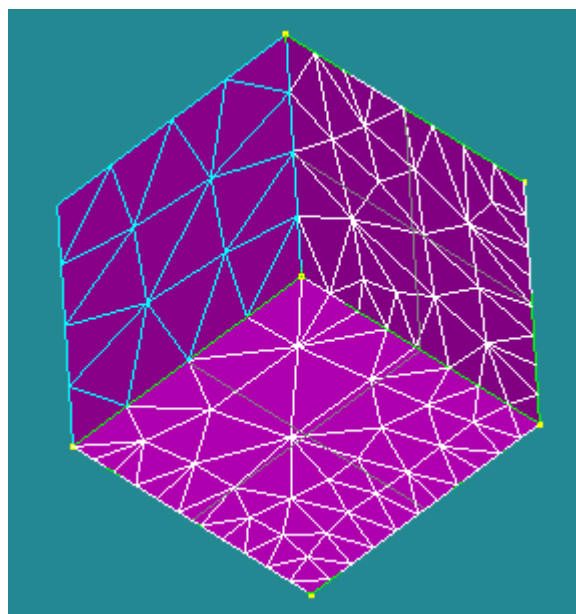
It is now possible to change the priority of the calculation of concurrent the submeshes during the mesh computation with the help of the **Change submesh priority** operation. This functionality is also available in the *Preview mesh* operation. For batch mode, the operation can be invoked using `GetMeshOrder()/SetMeshOrder()` commands. The python command `GetMeshOrder()` returns a list of detected concurrent submeshes for the given `Mesh` object. The python `SetMeshOrder()` command accepts lists of submesh objects (new order) as input argument.



Example: simple box is meshed with Netgen 2D-1D algorithm using a different number of edge segments for different faces: 4 segments for the first face, 8 segments for the second face and 12 segments for the third face.

Resulting mesh is shown in the figure below. It can be seen that the second and the third face inherit the number of elements on the shared edges from the first face. For these two edges, the sub-meshing order is substantial, since the number of splittings depends on the order in which sub-meshes are processed on a shared geometrical shapes (edges in this case).

With the new functionality, it is possible to change the order of the submeshing priority. For example, in the figure below, the face with 12 segments is processed first, then the face with 8 segments is calculated and finally face with 4 segments is processed, giving the result that is different from the default case (compare with the left figure).



Precision

The behavior of the spin box widgets in the Geometry, Mesh and Post-Pro modules have been improved. Now they do not accept floating-point input if its decimal precision is greater than a value specified in user preferences for this kind of quantities. Moreover, spin boxes for both integer and floating-point input guide the user an advice on how to input valid parameter values with a tool tip.

In connection with this improvement, the following capabilities have been implemented in the SALOME GUI module:

1. `QtDoubleSpinBox::validate()` uses `QDoubleValidator` to validate input in both standard (fixed-point) and scientific (exponential) notations. In case of scientific notation (`myPrecision < 0`), absolute value of `myPrecision` means the total number of significant digits in mantissa, while in case of standard notation `myPrecision` is the number of digits after the decimal point. Moreover, `validate()` does not accept the input in scientific notation whose exponent is outside the valid exponent range for double-precision numbers. If the optional `"strict_validity_check"` property is assigned to `QtDoubleSpinBox` instance, `validate()` method also blocks the input if it is out of range.

2. Additionally, `SalomeApp_IntSpinBox` and `SalomeApp_DoubleSpinBox` classes can optionally accept Notebook variable names (if `myAcceptNames` field is `true`), naming rules following that of Python variables. In case of invalid or incomplete input text, these classes show a tooltip with a valid range and - for `SalomeApp_DoubleSpinBox` class - an optional link to user preferences value that controls input field precision. The "link text" is passed to these classes by the "client" dialog boxes through dynamic `QObject` property with a name `"validity_tune_hint"`.

Tooltips can be turned off by `setShowTipOnValidate(false)` call.

Geometry, Mesh and Post-Pro modules have been updated to use improved versions of the `SalomeApp_IntSpinBox` or `SalomeApp_DoubleSpinBox` classes as spin box widgets, to have consistent ergonomics in all dialog boxes. User preferences for tuning precision of input fields have been added in these modules.

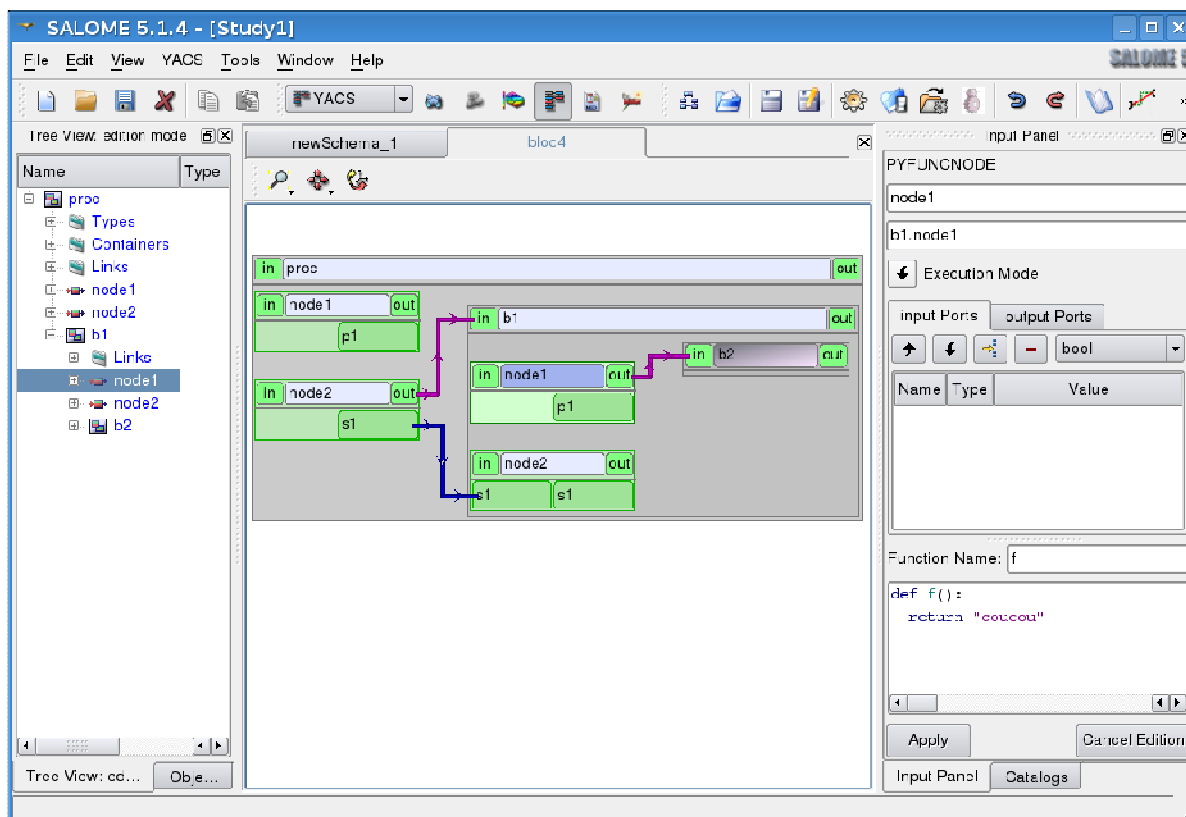
Note that in Post-Pro module preferences the former `"floating_point_precision"` parameter has been replaced with `"visual_data_precision"`, and it is necessary to adjust it to restore the previous behavior of VISU functionality that uses this parameter.

Note: For consistency of the application behavior, it is strongly recommended to avoid using `QSpinBox/QDoubleSpinBox` and `QtIntSpinBox/QtDoubleSpinBox` in a new functionality of standard SALOME modules and use `SalomeApp_IntSpinBox/SalomeApp_DoubleSpinBox` classes instead.

YACS improvements

- An Undo/Redo mechanism is now available in YACS GUI editor.
- In YACS GUI, the user can now collapse and expand blocks to hide or show internal nodes.
- In YACS GUI, size of nodes has been reduced. Control ports are now in the header and ports are smaller.
- Simplification of the panel associated to a node. Several useless items have been removed.
- Improve visibility of hierarchical blocks, by varying background color, depending on the level in the hierarchy.
- Make the node state colors a user preferences. They can now be changed in YACS preferences.
- Give a way to add nodes in a loop created with a single internal node. In the previous version, when a loop was created with a single internal node, it was not possible to add other nodes. The only way was to create another loop with an internal block and to copy the existing internal node. Now when the user tries to add a node to a loop with a single internal node, a dialog box proposes to put the internal node in a block and to add the new node in it. Moreover, a new item ("Put in Bloc") has been added to the context menu that can be used to put any node into a surrounding block.
- Give a way to change the value of input ports in run mode. This was not possible before without editing the schema.

- Move the input panel tabs to the top of the panel. The picture below shows the new YACS GUI look with a collapsed block (“b2”), varying background color, reduced node size and undo/redo buttons in the toolbar.



- In edition mode, display the error log on a node by node mode. When a node is not valid, the context menu has a new entry (“Node Error Report”) that can be used to display the node error. Before, the error was only displayed in the schema log.
- Add user preferred directories in the import file dialog box. The quick directory list (from SALOME general preferences) is added to the “Recent Places” list of the file dialog box when importing schema.
- Add a way to erase data, by program, in the buffer of a datastream port. It was possible to set a maximum size to the datastream buffer but not to call a function to clean the buffer. Four functions have been added: CPEFFI, CPEFFT, CPFINI, CPFINT. CPEFFI and CPEFFT erase all data after an iteration number or a time step. CPFINI and CPFINT erase all data before an iteration number or a time step.
- Add a way to set environment variable when executing a standalone component. It is now possible to set environment variable for a standalone component by setting properties on the associated component instance. A properties editor is available in the component instance panel to edit those properties. The property name is the name of the environment variable and the value its value.
- Add a reference documentation for CALCIUM datastream ports.

Other Improvements

- **Hexahedron** meshing algorithm has been improved to generate 3D mesh from the 2D skin.
- Menu command *New Entity* → *Primitives* → *Face* has been renamed to *New Entity* → *Primitives* → *Rectangle* to avoid confusion with the *New Entity* → *Build* → *Face* operation.
- A new version of the `GetShapesOnCylinder()` function allowing to specify cylinder location: `GetShapesOnCylinderwithLocation()` has been implemented.
- *Suppress Faces* dialog has been redesigned:

- The object shape is directly selected now, its name is shown in the corresponding field.
 - Now it is possible to select sub-shapes and groups in the Object Browser for the list of faces.
 - Option to keep sub-shapes and colors from the initial shape is available now.
- The *Second Shape* field of the *Create Group* dialog box of the Geometry module has been moved to the *Main Shape selection restriction* sub-panel, under the list of radio buttons. It is active only if any of two last radio buttons is checked.
- A TUI function to get the first or the last vertex of an edge or a wire, depending on its direction (orientation), or to find the vertex by its index in the wire has been implemented. The numeration of vertices starts from 0.
- It is now possible possible to perform **Extrusion**, **Extrusion along Path** and **Revolution** operations on several base objects at once.
- *Measures* toolbar has been added to Geometry module interface.
- A new option *Add prefix to names of restored subshapes* (the `AddPrefix` Boolean argument for Python API) has been added to the **Restore Subshapes** functionality. This option allows switching on/off the prefixes addition to the names of restored sub-shapes. If this option is checked, the function automatically adds the prefix "from_" to the names of restored sub-shapes and the prefix "from_subshapes_of_" to the names of partially restored sub-shapes. By default, this option is set to *true* value for the non-regression purposes.
- It has become possible to take the orientation of edges into account in the **Filling** algorithm of the Geometry module.
- It has become possible to export several meshes into a single MED file or add their data to the existing MED file (if the file selected in the export dialog already exists, an additional dialog is shown to choose between add and overwrite. However, if some of the selected meshes have the same names, or the MED file, to which the selected meshes are exported in "add" mode, already contains some meshes with the same names as the selected ones – the result file becomes incorrect since the same-named meshes with different structures are written over each other with errors.
- **Keyboard free interaction style** previously implemented only in the VTK viewer has been introduced also in the OCC viewer. It can be turned on from the *View Operations* toolbar or set up by OCC Viewer 3D preferences. This interaction style allows processing all view transformations without using the keyboard (only by mouse) and performing selection in the viewer by pressing "S" key. By default, the rotation in this mode is performed by the left mouse button, panning – the by middle mouse button, zooming – by the left and the middle mouse buttons pressed simultaneously.
- New Python function `getXVersion()` returning the hexadecimal representation of the SALOME version number has been introduced in the `salome_version.py` Python module.
- The possibility to export meshes and fields to the VTK binary format has been enabled in the MEDMEM library (SALOME Med module).
- The Python script generated with the **Dump Python** functionality now provides information about the version of SALOME platform.
- The **Filling** operation has been improved to check if the wires specified in the input compound can be converted to B-spline curves (to produce a single edge for each wire). If the conversion is successful for all wires in the compound, the resulting edges are passed to the Filling algorithm.
- It has become possible to set the frequency of time-stamps used to generate an AVI file using the new *Time stamp frequency* spin-box in the *Animation* dialog box of the Post-Pro module. The default frequency is equal to 1. If this value is increased, then only the time-stamps with indices divisible by this value will be captured, and the frame rate will be proportionally decreased – it allows generating a less voluminous film of the same duration.
- The *Merge Nodes* and *Merge Elements* dialog boxes have been improved to allow working in *Automatic* and *Manual* modes. The automatic mode allows merging nodes and elements without displaying them in the viewer (for performance reasons).

- New P1P1 interpolator for 3D unstructured/unstructured meshes has been introduced in the INTERP_KERNEL package of the Med module.
- It has become possible to save groups with the same names but of different entity types into a single MED file.
- The operations for import and export of tables in the CSV format have been implemented in the Post-Pro module.
- It has become possible to re-compute a mesh built with the Hexotic meshing plug-in after the hypothesis modification.
- The function `FIELD<>::integral(SUPPORT*)`, which computes the integral of a non-nodal field or its part given by optional argument has been added in MEDMEM package of Med module.
- It has become possible to automatically add newly created nodes and elements to mesh as belonging to the specified group (or even to create a new group of added elements).
- The *Create Group* dialog box of the Geometry module has been improved to allow hiding of some entities selected by the user. This functionality simplifies the process of selection of the internal geometrical sub-shapes. New buttons *Show only selected*, *Hide selected* and *Show all subshapes* have been added to the *Create Group* dialog box.



BUG CORRECTIONS

GUI MODULE (IAPP)

Summary: EDF 1250 GUI: Some dialog windows lost their cross button

Change: The "x" close button missing in some dialogs in version 5.1.3 has been restored.

Summary: EDF GUI : Changes the font with `qtconfig`, no effect ?

Improvement: Changes of the application behavior (style, font, and palette) are applied to the SALOME desktop even if SALOME style is turned on in the preferences.

Summary: EDF 1304 ALL: Recently opened study moved

Change: Now it is checked if "Most Recently Used" files really exist in the location from where they have been opened.

Summary: EDF 1414: Reloading a study provokes SIGSEGV error

Changes: Since study name can be changed outside GUI (e.g. from Python scripts), its name in GUI should be updated properly.

Summary: EDF 1416 GUI: Problems in the Notebook and with `SetColor` when switching the locale

Changes: Fixed problem of dumping and saving / restoring of Notebook variables on non-C locale.

KERNEL MODULE

Summary: [CEA 392] management of `dlopen` errors

Improvement: SALOME component loading mechanism has been improved.

- An out string argument has been added to container methods (`load_component_Library` and `create_component_instance_env`) to report the reason when load or create fails (INTERFACE CHANGE).
- Now it is checked if the implementation is C++, python or executable based on intermediate errors (`dlopen`, `dlsym`, `import` ...).
- Some specific algorithms have been moved in separate methods.

Summary: A strange `cast (const char*) → bool` in `SALOME_ModuleCatalog_Handler.cxx`

Change: A couple of wrong warnings have been removed.

GEOM MODULE

Summary: EDF 995 GEOM : Closed Polyline with tangence

Change: Closed interpolation on a set of points is now correctly processed by Python dump.

Summary: [CEA] Error with partition tool

Change: A problem with partition algorithm has been fixed.

<p><i>Summary:</i> EDF 1171 GEOM : Check compound of blocks: crash</p> <p>Change: "Remove Extra Edges" functionality has been improved to avoid corrupted results on compounds.</p>
<p><i>Summary:</i> EDF 1191 GEOM : Creation of hexa block from two faces</p> <p>Change: A warning is now shown at the attempt to add a sub-object under an unpublished parent object. Face, polyline and block creation functionality has been corrected.</p>
<p><i>Summary:</i> 20599: EDF 1187 GEOM: Creation of a quadrangle face from 2 edges: SIGSEGV</p> <p>Change: The algorithm of quadrangle faces creation from 2 edges has been improved</p>
<p><i>Summary:</i> EDF 1183 GEOM: Bounding values VS point coordinates</p> <p>Improvement: The inherent limitation on Bounding box functionality has been explained in the documentation.</p>
<p><i>Summary:</i> EDF 1188 GEOM: MakePartition fails on 1st try but succeeds on 2nd try</p> <p>Change: A problem in Partition algorithm has been fixed</p>
<p><i>Summary:</i> EDF 1172 GEOM : Result of Boolean operations, a null compound is not a correct result</p> <p>Improvement: The error message warning that the Boolean operation result is empty has been implemented.</p>
<p><i>Summary:</i> EDF 1200 GEOM: the discretisation of the circles in the OCC viewer is too rough</p> <p>New Feature: It has become possible to change the deflection coefficient of any shape, displayed in the OCC viewer using "Deflection coefficient" popup-menu item. It is also possible to change the default deflection for all new shapes in general preferences of the Geometry module.</p>
<p><i>Summary:</i> [CEA 386] GetType() function</p> <p>Improvement: The documentation about object types has been updated with their descriptions. A function ShapeIdToType() that converts the type of the geometry object from integer ID to the string value has been added to the geompyDC.py.</p>
<p><i>Summary:</i> EDF 1241 GEOM: Propagate: impossible to select a face in GUI</p> <p>Change: It has become possible to select not only solids and compounds, but also faces and shells in the Propagate dialog of Geometry.</p>
<p><i>Summary:</i> EDF 1239 GEOM: Change Orientation does not change the orientation of edges</p> <p>Improvement: Filling operation has been improved to take into account the orientation of edges (earlier only the orientation of curves was taken into account).</p>
<p><i>Summary:</i> EDF 1238 GEOM: Missing equivalent of 2nd constructor of LCS in TUI</p> <p>Improvement: Python implementation of creating local coordinate systems by Shape and by Point & 2 Vectors has been corrected.</p>
<p><i>Summary:</i> EDF 883 GEOM: MakeGlueFaces produces an invalid shape</p> <p>Improvement: Glue Faces functionality has been improved to avoid producing bad shapes.</p>

Summary: EDF 1212 GEOM: Partition operation creates vertices which causes mesh computation to fail with Netgen

Improvement: NETGEN meshing algorithm has been improved to take into account the internal geometrical vertices, which do not belong to the boundaries of geometrical faces.

Summary: EDF 1219 GEOM: `MinDistance` gives 0 instead of 20.88

Change: Detection of Minimum Distance value for a face shape created as a part of the torus shape has been improved.

Summary: EDF 1268 GEOM: Bad result of `GetInPlace` on a partitioned elliptic face

Change: Documentation on `Get Shapes on Shape` functionality has been improved to provide information about existing limitations.

Summary: EDF 1263 GEOM: `Suppress faces` does not remove faces and adds some more

Improvement: `Suppress Faces` functionality behavior has been changed for a compound argument: now the selected faces are removed from all its shapes and then `GlueFaces` is performed with the result.

Summary: EDF GEOM : `Makefile.in` in CVS

Change: Obsolete packages `GEOMDS`, `PARTITION` and `NMTA` have been removed.

Summary: Disparition of `LineEditReturnPressed` in `GroupGUI_GroupDlg.cxx`

Changes: The obsolete function `LineEditReturnPressed()` has been removed to avoid unresolved symbols problem.

Summary: EDF 1232 GEOM : Pipe construction dialog box, impossible to select a group or a compound

Improvement: It has become possible to select the Compound as Base Object for pipe construction.

Summary: EDF 1302 GEOM: Fillet 1D anomaly

Improvement: The Wire Creation algorithm has been corrected to allow creating only one wire from a chain of edges that can be combined into a wire with manifold topology.

Summary: EDF 1313 GEOM : Fillet 1D can crash the application and the `Makewire` tolerance is not dumped

Change: Crash problems with Fillet 1D functionality have been eliminated.

Summary: EDF 1280 GEOM: Function 'Add Point on Edge' (in Repair menu)

Changes: The documentation of "Add point on Edge" function has been improved to clarify the difference between "by parameter" and "by length" operation modes.

Summary: EDF 1320 GEOM: Script regression, bad result of GEOM operations

Improvement: The Pipe creation algorithm has been improved to avoid changing the base object after the operation.

Summary: EDF 1340 GEOM : Dump Study & Notebook for the 3D Sketcher

Improvement: It is now possible to use Notebook variables with 3D sketcher functionality in GUI and batch mode. Additionally, "X", "Y" and "Z" spin boxes of the dialog box now have zero values indicated by default.

Summary: EDF 1342 GEOM : Group of points from parent objects are not taken into account

Change: "Set presentation parameters and subshapes from arguments" option has been corrected to restore the vertex subshapes.

Summary: EDF 1350 GEOM : Partition issue

Change: A problem in Partition algorithm has been fixed.

Summary: EDF 1371 GEOM : Selection problem with the sketcher

Changes: A problem with direct selection of vertices in the "2D Sketcher" dialog box (impossibility to select some points) has been fixed.

Summary: EDF 1398 GEOM: Ergonomics of Partition GUI window

Change: The behavior of the Partition dialog has been improved: now it is not emptied, when the user clicks "Apply" button.

Summary: EDF 1405 GEOM: Impossible to create point in sketcher with LANG=fr_FR.UTF-8

Change: The problem with using a non-C locale in 2D Sketcher dialog box has been fixed.

Summary: EDF GEOM: A former script doesn't work anymore (regression).

Changes: Changed behaviour of `geompy.ProcessShape()` method in case of good (with no error) passed shape: now it doesn't raise an exception, but returns the initial shape.

Summary: EDF GEOM: Compound of vertices

Changes:

- Fixed an error in `GetInPlace()` (leading to the error of "Set presentation parameters and subshapes from arguments").
- Improved "Set presentation parameters and subshapes from arguments": point markers are restored now.
- Contextual menu of vertices improved ("Point marker" can be assigned to any vertex now, not only to "GEOM_POINT").

SMESH MODULE

Summary: EDF 661 SMESH : to be able to mesh a prismatic shape as a revolution of meshed surface

Change: Fixed problem with the documentation of the Quadrangle Parameters hypothesis.

Summary: EDF 681 SMESH – Pre-evaluation of the number of elements before mesh

Change: Additional fixes have been implemented to improve the results of Evaluate functionality.

Summary: EDF SMESH 926 : Quadratic conversion of BLSURF mesh

Changes: Fixed additional problems of the quadratic/non-quadratic mesh conversion.

Summary: EDF 1190 SMESH: Display performance. 80 seconds for 52000 cells.

Change: The problem with long visualization of a SMESH mesh created via a python script without activating the SMESH module has been fixed.

Summary: EDF 1206 SMESH: SMESH group edition and object browser highlight

Improvement: Highlighting of group editing in the viewer has been corrected in SMESH Edit Group dialog.

Summary: EDF 1207 SMESH: DoubleNode: impossible to duplicate edges in a 3D mesh

Change: PYTHON interface to `SMESH_MeshEditor::DoubleNodeElemGroupInRegion()` function has been fixed to correctly duplicate edges in a 3D mesh.

Summary: EDF 1208 SMESH: problems with `DoubleNodeElemGroup*InRegion`

Change: It has become possible to use a geometrical face as a bounding box passed to `SMESH_MeshEditor::DoubleNodeElemGroupInRegion()` function.

Summary: EDF SMESH: Union list regression ?

Improvement: `UnionList` function now doesn't raise an exception in case of empty second argument.

Summary: EDF 1194 SMESH: Merge nodes or elements: impossible to select a mesh if it is not displayed

Change: It has become possible to select the whole mesh after closing of "Merge Elements" dialog by "Esc" button – earlier it was possible to select only its elements. Additionally the mesh now can be selected in the Object browser even if it was not previously displayed in the viewer. On the other hand it is now impossible to add an empty group of elements by clicking "Add" button without selecting any elements in the viewer.

Summary: EDF 1209 SMESH: Load script/execfile add the objects in the wrong study

Change: Python examples have been improved to run properly in different studies.

Summary: EDF 1223 SMESH: Error message of `RadialQuadrangle` algorithm not precise.

Change: If the `RadialQuadrangle` algorithm is applied on a wrong face, the error message now warns the user that "the face must be a full circle or a part of circle".

Summary: EDF 1225 SMESH: Missing options in Color/Size menu

Improvement: Colors/Size dialog has been extended with controls for parameter setting of 0D elements visualization and "Orientation of faces" features.

Summary: EDF 883 SMESH: A face is badly meshed with BLSurf – Netgen1D2D is OK

Improvement: BLSURF meshing algorithm has been modified to create nodes on geometrical vertices and edges at their 3d representation instead of places of their projections on the surface of a face.

Summary: EDF 1244 SMESH: Bad display of tetras

Change: Proper visualization of the outline actor has been restored.

Summary: EDF 1252 SMESH: Bad cell created by Radial prism in the centre of torus

Change: "Extrusion 3D" meshing algorithm has been fixed to work correctly with shapes having problematic B-spline surfaces.

Summary: EDF 1260 SMESH: Non uniform cells with Fixed Points hypothesis

Improvement: The problem with invalid work of Fixed Points hypothesis on edges with non-uniform parameterization of curves has been fixed.

Summary: EDF 1222 SMESH: 3D mesh from a skin mesh and with volume cells

Improvement: NETGEN and GHS3D plugins have been fixed to avoid creation of tetrahedrons at the places where 3D elements are already present. NETGEN plugin has been improved to correctly work if an input 2D mesh has several sub-domains with boundary of quadrangles.

Additionally, "Tetrahedron (NETGEN)" meshing algorithm has been improved to be able to mesh 3D geometrical shapes with "internal" faces, which are not shape boundary.

Summary: EDF SMESH : Export STL always in binary

Change: The problem with mesh export to STL ASCII format from GUI has been fixed.

Summary: EDF 1248 SMESH: Sub-mesh on group of face not taken into account

Changes: Some problems with sub-meshes on geometrical groups have been fixed.

Summary: EDF 1234 SMESH: Quadrangle preference and Netgen

Change: It has become to use Quadrangle Preference hypothesis with NETGEN 2D meshing algorithm.

Summary: EDF SMESH : GetName Method of smeshDC.py

Improvements: The function GetName() of SMESH Python API has been fixed to work correctly in a multi-study environment.

Summary: EDF 1261 SMESH: Radial Quadrangle algorithm produces "curved" segments instead of straight ones

Improvement: The problem with invalid work of Radial Quadrangle algorithm on edges with non-uniform parameterization of curve has been fixed.

Summary: EDF 1233 SMESH : Crash/bad behavior of 'Convert linear Quadratic with Medium Nodes on Geometry' feature with BLSurf/Ghs3D

Change: The problems of crashes and bad results of "Convert to quadratic" operation on geometries with periodic surfaces have been eliminated.

Summary: QWT_LIBS in SMESH_SRC/src/StdMeshersGUI/Makefile.am

Change: Missing linkage options for Qwt have been added to the StdMeshersGUI library.

Summary: EDF 1276 SMESH : add a Method RadialQuadrangle to the Mesh class in smeshDC.py

Changes: Fixed misprint in the smeshDC.py module for algo type enumeration.

Summary: EDF 1305 SMESH: Create 2D mesh from 3D

Change: The problem with taking into account the existing 2d elements by "2D mesh from 3D" functionality has been solved.

Summary: [CEA] Computing mesh fails

Change: Regression of Quadrangle 2D meshing algorithm in case of quasi quadrilateral face has been fixed.

Summary: EDF 1310 SMESH: Mesh to pass through point operation crashes Salome

Improvement: The problem with unlimited recursion of a mesh with many coincided nodes has been solved.

<p><i>Summary:</i> EDF 1322 SMESH : Quadratic/Linear conversion with Medium Nodes on Geometry issue</p> <p>Change: The problem with mesh conversion to quadratic has been fixed.</p>
<p><i>Summary:</i> EDF 1326 SMESH : Documentation missing</p> <p>Change: "Orientation of faces" visualization feature has been described in more detail in SMESH documentation.</p>
<p><i>Summary:</i> EDF 1339 SMESH : Crash after enabling "Auto Color" feature in attached MED file</p> <p>Change: It has become possible to publish in the study groups with 0D elements imported from a MED file. Additionally, auto color feature now properly works with groups of 0D-elements.</p>
<p><i>Summary:</i> EDF 1336 SMESH : Transformation of a 0D element</p> <p>Change: 0D elements are now properly taken into account during transformation operations.</p>
<p><i>Summary:</i> [CEA] Smesh script fails</p> <p>Change: A problem with script crash has been fixed.</p>
<p><i>Summary:</i> EDF 1343 SMESH : Segmentation violation during compound of Mesh with 0D elements</p> <p>Change: SIGSEGV error with creation of a compound from mesh containing 0D element(s) has been fixed.</p>
<p><i>Summary:</i> EDF 1369 SMESH : Double elements generated</p> <p>Improvement: "Number of segments" meshing hypothesis has been improved to properly work on reversed edges with "Distribution with table density" type.</p>
<p><i>Summary:</i> EDF 1374 : SMESH Projection 2D + Extrusion 3D issue</p> <p>Improvement: "3D Extrusion" meshing algorithm has been improved to work with prismatic shapes having quasi quadrilateral side faces.</p>
<p><i>Summary:</i> EDF 1378 SMESH : Selection problem with symmetry + translation</p> <p>Change: The problem with selection of a newly loaded mesh in some dialogs has been corrected.</p>
<p><i>Summary:</i> [CEA] Projection 2D fails</p> <p>Change: "Projection 2D" meshing algorithm has been fixed to work correctly on geometrical faces with non-uniform parameterization of edges.</p>
<p><i>Summary:</i> [CEA] V5_1_4_BR - script fails</p> <p>Changes: Regression of MEFISTO meshing algorithm on a spherical geometry has been fixed</p>
<p><i>Summary:</i> [CEA] Problem with ijk algo</p> <p>Change: "Hexahedron (i,j,k)" meshing algorithm has been fixed to work correctly on a skin mesh (without geometry)</p>
<p><i>Summary:</i> EDF SMESH : Dump of attached script is not running.</p> <p>Change: Dump Python Salome functionality has been improved.</p>

Summary: EDF 1404 SMESH: Save operation unavailable after a group creation or merge nodes/elements

Improvement: An auxiliary method that sets the study as modified and updates the state of GUI actions is called by the click of Apply button in the dialogs that modify the study, which allows saving the study after each performed operation.

Summary: [CEA 403] SMESH hypotheses GUI destruction

Change: A problem with the hypothesis creation/editing dialog box still open when Mesh module is deactivated by the user has been fixed.

Summary: EDF 1409 SMESH: Suppression of algorithms causes Salome to crash

Change: A crash occurring at the attempt to delete several objects selected continuously from newer to older, i.e. in reverse order, in GEOM, SMESH and other modules has been fixed.

Summary: EDF 1251 SMESH: Pattern 3D mapping

Improvement: The problem with using non-C locale in the Pattern Mapping operation has been fixed. Additionally, the documentation and sample Python scripts for 3D patterns have been added.

Summary: EDF 1411 SMESH: Distribution with analytical density crashes Salome

Change: A crash occurring if the analytical function of nodes distribution yields an infinite value has been fixed.

Summary: [CEA] Script fails

Changes: The changes done in `geompy` in the frames of issue 0020052 have been reflected in SMESH sample scripts.

Summary: EDF 1412 SMESH: Conversion linear to quadratic with BLSURF crashes Salome or doesn't work

Changes: In SMESH, conversion of mesh to quadratic has been fixed for the case where the mesher sets nodes on geometrical edges instead of vertices.

Summary: EDF 1415: an example is missing in SMESH documentation

Changes: Integrated documentation example for the TUI creation of 0D Element.

Summary: EDF 1429 SMESH: Remove elements using "Set Filters" fails

Changes: fixed bug with entering big number of the nodes / elements identifiers in the "Remove Nodes", "Remove Elements", "Translation", "Rotation", "Symmetry", "Extrusion", "Extrusion Along a Path", "Mesh Element Info" dialog boxes - removed default maximum size limit on the input string.

VISU MODULE

Summary: EDF PAL 569: Cut plane on Deformed

Change: Sweep functionality has become unavailable for presentations built on fields.

Summary: EDF 885 VISU: Unchecking Filter by Scalar does not remove white marks on the scalar bar

Improvement: Lookup table of the scalar bar actor is now rebuilt when the scalar filtering is turned off.

Summary: EDF VISU SMESH: the dump view command does not remove popup menu

New Feature: Dump View functionality has been improved for VTK and OCC views. It has become possible to dump contents of these views even if they are overlapped or off-screen.

Summary: [CEA 390] VISU2D

Changes: The *.x1s extension has been removed from the list of available file filters in the "Import Table" dialog box.

Summary: EDF 1292 VISU: SURFACEFRAME on a scalar map using a python script

Change: It has become possible to set surface frame as the representation type for Post-Pro presentations via python console.

Summary: [CEA 395] Problem of representation of a group lying on a family lying on nodes and cells

Improvement: The groups on families defined both on node and cell entities and thus duplicated in VISU data structure are now referred to by two instances related to each type of entity.

Summary: EDF 1237 VISU : Crash Salome with Stream Lines

Improvement: The problem causing application crash during building of a Stream Lines presentation on fields containing very small scalar values has been fixed.

Summary: EDF 1308 VISU: Gauss point picking not possible when a mesh representation is displayed

Change: The problem with picking Gauss Points displayed together with mesh presentation has been solved.

Summary: [CEA] Visu example fails

Change: Some python scripts have been updated.

Summary: EDF 1347 VISU : Surface Representation during edition

Changes: The problem with incorrect representation mode of Deformed Shape presentation during its edition has been solved.

Summary: Impossible to create AVI file from animation (bad compilation of Qt ?)

Changes: `VISU_TimeAnimation::checkAVIMaker()` method has been slightly corrected. Since static `QImageWriter::supportedImageFormats()` method returns list of strings in lower case (at least, in Qt 4.x version), it is necessary to look for "jpeg" entry in this list during the check procedure, not "JPEG".

MED MODULE

Summary: [CEA 354] attribute initialization missing in copy constructor of `MEDMEM::MESH`

Improvement: Copy constructor of `MEDMEM::MESH` class has been improved to initialize all necessary class fields.

Summary: [CEA] V5.1.3 – Error during make check

Changes: Instability of `MEDMEMTest::test_remapper4()` test case on 32bit/64bit platforms has been fixed.

Summary: [CEA 373] MEDMEM_SUPPORT::blending const

Change: An argument of SUPPORT::blending(SUPPORT *) method has become constant.

Summary: [CEA] difference between two pointe.med

Change: The file pointe.med has been changed in SAMPLES repository.

Summary: [CEA 370] Problem of writing with VTKFieldDriver

Change The problem with incorrect writing of double fields by MEDMEM_VtkFieldDriver has been fixed.

Summary: [CEA 374] New Method in MEDMEM library

New Feature: New method buildSupportOnNode() has been added to the MEDMEM::SUPPORT class. This method creates a new SUPPORT of all nodes belonging to the elements of a given SUPPORT. The new method is also available via PYTHON interface of MEDMEM.

Summary: [CEA 371] VTK field driver : save many fields

Change: The incorrect work of MEDMEM_VtkFieldDriver::writeAppend() has been fixed for the case of writing many fields.

Summary: [CEA 378] Abort during MED mesh reading

Change: The problem with impossibility to read a structured mesh with families and fields from a med file has been fixed in MEDMEM.

Summary: [CEA 381] MEDMEM:REMAPPER : intersection error

Change: MEDMEM_REMAPPER has been fixed to work correctly with meshes containing polyhedrons and pentahedrons.

Summary: Wrong behavior with a structured field

Change: MEDwrapper and VISU have been fixed to support fields on sub-cells of a structured mesh.

Summary: [CEA 384] Problem with 1D no structured meshing

Change: In MEDMEM, the type of a field support entity read from a 1D field in a med file has been fixed

Summary: CEA 376] #define in MEDMEM (and elsewhere ?)

Improvement: Header files in MEDMEM, INTERPKERNEL and INTERPOLATION packages have been modified by replacing #define statements with constants or enumerations. Some of such constants and/or enumerations have been moved to the namespaces. This improvement has been implemented to decrease possible conflicts with other libraries.

Summary: EDF 1311 MED : Tui Doc mistake for SALOME_MED.MED.getFieldIterations function

Change: An error in the documentation of SALOME_MED.MED.getFieldIterations() function has been fixed.

Summary: [CEA] MEDMEM – getGroup function in python

Improvement: Missing getGroup(char* name) function has been added to the Python interface (SWIG) of the MEDMEM::MESH class.

Summary: [CEA 399] Memory corruption in MEDMEM::FIELD::integral in testField in MEDMEMCpptest

Change: Some problems in MEDMEM, detected with valgrind tool during MEDMEMCpptest/.libs/TestMEDMEM execution have been fixed.

Summary: [CEA] integration of cxx file

Change: A new version of MED_SRC/src/RENUMBER/renumbering.cxx has been integrated.

Summary: [CEA 401] error when using medsplitter

Change: MEDMEM::MESHING class has been fixed to enable creation of a 2D mesh in the 3D space.

Summary: [CEA] Failure with ParaMedMem

Change: The failure of TestParaMEDMEM executable has been fixed.

YACS MODULE

Summary: [CEA] Check of Qt in YACS throw ERROR if not found

Change: Qt has become mandatory for compilation with GUI and optional otherwise to avoid the YACS module compilation problem.

Summary: [CEA] Corba test fails

Change: SalomeContainer has been fixed to correctly read old XML schema files having only one long value for nb_component_nodes property.

Summary: No way to initialize a file port in YACS GUI. It is now possible to initialize a file port with a file name.

Summary: Node name not correctly updated in containers tree on rename operation.

Summary: YACS GUI can't handle dataflow and datastream ports with same name for the same node.

Summary: YACS GUI crashes when an exception is thrown in a component.

Summary: Deadlocks can occur with datastream coupling on SALOME shutdown
The shutdown of SALOME has been improved (mainly shutdown of connection manager).

Summary: Deadlock with datastream coupling if nodes number is larger than the limit number given by YACS_MAX_THREADS. Now this situation is detected and a warning printed.

Summary: Crash when editing a GEOM object in YACS GUI (Python lock problem).

Summary: Node names can be larger than the size node. They are now truncated to the node size. The tool tip gives the full name.

Summary: Condition id in a switch node can't be edited.

Summary: The edition of a python script was lost if the apply button was not clicked.

Summary: Several problems, related to container edition after deleting a node.

Summary: Crash when switching from a study to another.

Summary: Copy, paste, cut, delete should not be available in run mode. These actions have been removed in run mode.

Summary: COMPONENT_INSTANCE_NEW not translated.

Summary: Problem with tree view on selection (erratic jump).

Summary: Crash in YACS GUI when loops have a large number of turns. There was a problem with concurrent access that has been solved with a mutex to guarantee exclusive access.

Summary: When nodes are created they hide existing nodes.

Summary: Problem with accents and other special characters in node names and schema names. Accents are no more accepted.

NETGENPLUGIN MODULE

Summary: EDF 1329 NETGENPLUGIN : Create Group from Geometry issue

Change: The problem with associating a node generated by Netgen 1D-2D meshing algorithm with a wrong geometrical vertex has been fixed.

Summary: EDF 1330 NETGENPLUGIN: Netgen cannot mesh a face partitioned with 3 edges

Change: NETGEN plugin has been fixed to take into account that the same internal vertex can belong twice to a geometrical face.

HEXOTICPLUGIN MODULE

Summary: [CEA] Can't decrease hexotic parameters

Change: It has become possible to decrease parameters for Hexotic mesh creation.

GHS3DPLUGIN MODULE

Summary: EDF 870 SMESH: Mesh generated by Netgen not usable by GHS3D

Change: GHS3D plugin has been improved to mesh geometry with degenerated edges.

Summary: [CEA] Mesh computation fails

Change: GHS3Dplugin has been fixed to avoid crash.

Summary: EDF GHS3DPLUGIN: shapeToMesh when creating 3D mesh from 2D mesh

Improvement: It has become possible to find the point position in a closed 2D mesh using SMESH_ElementSearcher function.

BLSURFPLUGIN MODULE

Summary: EDF 1264 BLSURFPLUGIN: Crash of Salome during mesh computation, locale issue

Change: BLSURFPlugin_BLSURF::Compute() method has been wrapped by Kernel_Utils::Localizer to avoid problem with locales..

Summary: [CEA 398] Problem with BLSurf Plugin

Change: The Gradation Spin box value is now properly taken into account.

YACSGEN TOOL

Summary: Do not include first argument compo in a service that has no datastream ports

Summary: Add a new configuration parameter (named "resources") to set the resources catalog of an application.

Summary: It is now possible to use some data types defined in SALOME modules when defining dataflow ports (mainly GEOM_Object and MED objects).

Summary: It is now possible with the two parameters "inheritedclass" and "compodefs" to derive a component from a C++ or a Python class.

INSTALLATION PROCEDURE

Summary: [CEA] InstallWizard 5.1.4 rc1

Changes: Fixed problem with JOBMANAGER installation script.

BUILD PROCEDURE

Summary: [CEA 391] Bug during HXX2SALOME compo compilation when one directory installation

Improvement: Configure script check_GUI.m4 has been synchronized in all modules to eliminate the problem with collision of different versions of check_GUI.m4 script.

Summary: Duplicate check_*.m4 between modules

Improvement: The script build_configure has been updated for some modules that require check_*.m4 scripts located in GUI module (check_opengl.m4, check_qt.m4, etc.).
Now these modules get the scripts from the GUI installation directory since GUI_ROOT_DIR environment variable is defined correctly.

Summary: [CEA] OneDir compilation fails

Change: The conflict between same-named header files in the MED and YACS modules has been fixed.



CASCADE 6.3 SERVICE PACK 9 BUG CORRECTIONS

Summary: New define has been added for include "unistd.h" in lex.*.c of StepFiles and ExprIntr packages for windows platform

Summary: New define OCC_VERSION_SERVICEPACK has been added to Standard_Version.hxx
Sample for sp9 : #define OCC_VERSION_SERVICEPACK 9

Summary: OCC21171 Problem with vfit of vertex and edge

Referenced by "0020668: [CEA] FitAll and points" and "20775: EDF 1327 GEOM: FitAll on lines do not work".

Improvement: The Fit All operation for OCC viewer has been improved to take into account points and lines which previously have been ignored (as "infinite" objects).

Summary: OCC21499 ShapeAnalysis_FreeBounds::ConnectEdgesToWires() builds wrong wires

Referenced by "0020596: EDF 1115 GEOM: Creation of a face with vectors, lines and arcs".
The problem leading to incorrect work of MakeFace in some cases has been fixed.

Summary: OCC21534 Need correction of IntTools_Context (attached) for SALOME partition algorithm

Referenced by "20603: EDF 1188 GEOM: MakePartition fails on 1st try but succeeds on 2nd try".
Fixed a problem in Partition algorithm, reproduced on some platforms.

Summary: OCC21548 Problem with cut operation. Appendix for Mantis 0020621 issue

Referenced by "0020621: EDF 1211 GEOM: result of cut operation different depending on order".
A problem in Boolean operations has been solved.

Summary: OCC21572 The bug is appendix to the SALOME Bug 0020642

Referenced by "20642: EDF 1227 GEOM: Partition fail".
Changes: A problem with the Partition algorithm has been fixed.

Summary: OCC21588 Appendix to the SALOME bug: 0020683

Referenced by "20683: EDF 1249 GEOM: Sometimes Partition adds crosses on a solid that has to be meshed with prisms".
Changes: A problem with the Partition algorithm has been fixed.

Summary: OCC21603 The bug is appendix to the SALOME Bug 0020707

Referenced by "20707: EDF 1224 GEOM: Result of MakeCommon is not valid".
A problem in Boolean operations has been solved.

Summary: OCC21609 Bad result after re-writing to IGES.

Referenced by "0020726: EDF 1259 GEOM : Bug d'export IGES 5.1".
Improvement: The problem with reading some cases of IGES entities with wires containing seam edges has been fixed.

Summary: OCC21616 ShapeAnalysis_FreeBounds problem

Referenced by "0020729: EDF 1256 GEOM: Partition between 2 contact surfaces".
Improvement: Internal edges are now taken into account for free boundary detection.

Summary: OCC21618 Appendix to the SALOME bug: 0020730

Referenced by "20730: EDF 1282 GEOM : MakeCut issue" and "EDF 1284 GEOM: MakeCommon issue".
Change: Some algorithms of Boolean operations have been corrected.

Summary: OCC21638 The bug is appendix to the Salome Bug 0020773, 0020774

Referenced by "20773: EDF 1324 GEOM: MakeCut Issue" and "20774: EDF 1325 GEOM: MakeCommon Issue".
Change: Boolean operations algorithms have been corrected.

Summary: OCC21645 Slow and bad triangulation on attached shape.

Referenced by "0020718: EDF 1231 GEOM : Very very long display with deflection".
Improvement: The problem with triangulation of small faces has been fixed.

SUPPORTED LINUX DISTRIBUTIONS AND PRE-REQUISITES

SALOME 5.1.4 supports Debian 3.1 Sarge, Debian 4.0 Etch 32bit and 64bit, Mandriva 2006 32bit and 64bit, Mandriva 2008 32bit and 64bit. SALOME 5.1.4 version has been mainly tested with the following pre-requisite list on Mandriva 2008 32bit and Debian 4.0 Etch 64bit platforms.

SALOME 5.1.4 comes with the same prerequisites versions on all supported platforms (with some exceptions). The table below lists the versions of the products used by SALOME platform. Other versions of the products can also work but it is not guaranteed.

NOTE: For some platforms Salome uses prerequisites with patches like in RPM and defines specific keys. If you compile products without Install Wizard we strongly recommend you to check compilation keys using shell files located in config_files folder of the Installation Procedure.

	Version	GUI (IAPP)	KERNEL	GEOM	SMESH	VISU	MED	YACS	NETGENPLUGIN	GHS3DPLUGIN	GHS3DPRPLPLUGIN	BLSURFPLUGIN	HexoticPLUGIN	RANDOMIZER	SIERPINSKY	PYCALCULATOR	COMPONENT	CALCULATOR	HELLO	PYHELLO	LIGHT	PYLIGHT	MULTIPR	xdata	HXX2SALOME
gcc*	3.3.5**	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
automake*	1.9**	X	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X	X	X	X	X
autoconf*	2.59**	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
libtool*	1.5.6**	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
GNU make*	3.80**	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Python	2.4.4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Qt	4.5.2	X		X	X	X	X	X	X	X	X	X	X		X		X	X	X	X	X		X	X	X
Sip	4.8.2	X															X								
PyQt	4.5.4	X															X					X		X	
Boost	1.40.0***	X	X	X	X	X	X		X	X	X	X	X		X			X	X				X		
Swig	1.3.40	X	X	X	X	X	X	X	X	X	X	X	X		X		X	X					X		
OpenCASCADE Technology	6.3 sp9	X		X	X	X	X	X	X	X	X	X	X		X		X	X	X		X		X		
Qwt	5.2.0	X															X								
QScintilla	2.4							X																	
OmniORB	4.1.4																								
OmniORBpy	3.4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	
omniNotify	2.1																								
Hdf5	1.6.9	X	X	X	X	X	X		X	X	X	X	X		X		X	X			X		X		
Med	2.3.6				X	X	X		X		X				X	X	X	X					X		
Vtk	5.0.4	X		X	X	X	X		X	X	X	X	X		X		X				X	X		X	
numpy	1.3.0		X																						
lapack	3.2****		X																						
Graphviz	2.24.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X				
Doxygen	1.6.1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X				
NETGEN	4.5								X																
docutils	0.6.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X				
metis	4.0						X																		
scotch	4.0						X																		
libxml2	2.6.27	X	X				X	X																	
blsurf	2.8											X													
TetMesh-GHS3D	4.1									X	X														
TcITk	8.4.14																								
Sphinx	0.6.3							X																	
Expat	2.0.1							X																	
libBatch	1.1.0		X																						

*) Not included into SALOME Installation procedure

**) Minimal required version

**) 1.37.0 for Debian 3.1 Sarge since version 1.40.0 is not compatible with gcc 3.3 used at Debian Sarge

****) 3.1.1 for Debian 3.1 Sarge since version 3.2 is not compatible with gcc 3.3 used at Debian Sarge

SALOME 5.1.4 depends of a number of products for run time execution, others are necessary only for compilation or generation of development documentation (like doxygen for example). Below there is a list of mandatory and optional products.

Software Requirements

	Compilation and Development		Execution		Remarks
	Mandatory	Optional	Mandatory	Optional	
gcc	X		X		
Automake	X				
Autoconf	X				
libtool	X				
GNU make	X				
Tcltk					for OCCT compilation from source files only
Python	X		X		
Qt	X		X		
sip	X				
PyQt	X		X		
Boost	X		X		
Swig	X				
OpenCASCADE Technology	X		X		
Qwt	X		X		
QScintilla		X		X	
OmniORB	X		X		
Hdf	X		X		
Med	X		X		
Vtk	X		X		
numpy		X			
Graphviz		X			
Doxygen		X			
NETGEN	X				for NETGENPLUGIN mesh plug-in only
docutils		X			for KERNEL and YACS documentation only
cppunit		X			
mpi		X		X	required only if used at compilation step
openpbs		X		X	required only if used at compilation step
Lsf		X		X	required only if used at compilation step
metis	X		X		
scotch	X		X		
libxml2	X				
blsurf	X		X		for BLSURFPLUGIN mesh plug-in only
TetMesh-GHS3D			X		for GHS3DPLUGIN mesh plug-in only
Sphinx		X			for YACS documentation only
Expat	X				For YACS only
libBatch		X			

HOW TO INSTALL AND BUILD SALOME

Please follow README file from Installation Wizard for processing correctly installation of SALOME and all prerequisites.

If you would like to compile SALOME from scratch, please use `build.csh` or `build.sh` script delivered with the Installation Wizard. Call "`build.csh -h`" to see available options of this script.

SALOME SYSTEM REQUIREMENTS

Minimal Configuration:

- Processor: Pentium IV.
- 512 Mb RAM.
- Hard Drive Space: 1.5 Gb.
- Video card 64mb.
- CD/DVD

Optimal Configuration:

- Processor: Dual Core.
- 2 Gb RAM.
- Hard Drive Space: 5Gb.
- 2Gb Swap.
- Video card 128mb.
- CD/DVD

HOW TO GET THE VERSION AND PRE-REQUISITES

SALOME 5.1.4 pre-compiled binaries for Linux Debian 3.1 Sarge, Mandriva 2006 (32bit and 64bit), Mandriva 2008 (32bit and 64bit), Debian 4.0 Etch (32bit and 64bit) can be retrieved from the <http://www.salome-platform.org> site.

Besides it is possible to install Debian 3.1 Sarge Salome universal binaries on the platforms that are not supported officially.

The SALOME Installation procedure includes SALOME modules sources, and it is possible to build sources from scratch using `build.sh` or `build.sch` script coming with installation procedure.

There are two patches on **NETGEN** which are placed inside NETGENPLUGIN modules sources. The first patch file is used for all 32 bit platforms; the second patch file is an addition to the first one and should be applied only for 64bit platforms.

During the compilation on NETGEN from sources by SALOME Installation Wizard, the patches are applied automatically to the standard NETGEN distribution. You can download NETGEN 4.5 from its official site using the following link: <http://www.hpfem.jku.at/netgen>.

All other pre-requisites can 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

- The following modules have not been migrated to Qt series 4 and thus are not included into SALOME 5.1.4 release: FILTER and SUPERV.
- The following limitations refer to BLSURF plug-in:
 - Mesh contains inverted elements, if it is based on a shape, consisting of more than one face (box, cone, torus...) and if the option "Allow Quadrangles (Test)" has been checked before computation.
 - SIGFPE exception is raised after trying to compute a mesh based on a box with "Patch independent" option checked.
 - It has been found out that BLSURF algorithm can't be used as a local algorithm (on sub-meshes) and as a provider of low-level mesh for some 3D algorithms because BLSURF mesher (and, consequently, the plug-in) does not provide information on node parameters on edges (U) and faces (U, V). For example, the following combinations are impossible:
 - global MEFISTO or Quadrangle(mapping) + local BLSURF;
 - BLSURF + Projection 2D from faces meshed by BLSURF;
 - local BLSURF + Extrusion 3D.
- Sometimes regression test bases give unstable results; in this case the testing should be restarted.
- A native VTK can be used only after manual recompilation with the GL2PS component.
- NETGEN 1D-2D and 1D-2D-3D algorithm do not require definition of 2D and 1D algorithms and hypotheses for both mesh and sub-mesh. 2D and 1D algorithms and hypotheses defined with NETGEN 1D-2D or 1D-2D-3D algorithm will be ignored during calculation.
- SALOME supports reading of documents from earlier versions but the documents created in the new version may not open in earlier ones.
- If SALOME modules are not installed in a single folder, SALOME may not work in the CSH shell since the environment variables are too long by default. In this case, it is suggested to use SH or to install all modules in the same folder.
- During the compilation of OCT 6.x by makefiles on a station with NVIDIA video card you can experience problems because the installation procedure of NVIDIA video driver removes library libGL.so included in package libMesaGL from directory /usr/X11R6/lib and places this library libGL.so in directory /usr/lib. However, libtool expects to find the library in directory /usr/X11R6/lib, which causes compilation failure (See /usr/X11R6/lib/libGLU.la). We suggest making symbolic links in that case: `ln -s /usr/lib/libGL.so /usr/X11R6/lib/libGL.so ln -s /usr/lib/libGL.la /usr/X11R6/lib/libGL.la`.
- VISU module does not support timestamps defined on the same field but on different meshes
- Stream lines presentation can not be built on some MED fields due to limitations in VTK.
- MEFISTO algorithm sometimes produces different results on different platforms.
- In some cases the number of triangles generated by MEFISTO may be different at each attempt of building the mesh.