

# SALOME 3.2.3pre1

## Maintenance release announcement

November 2006

### General information

OPEN CASCADE is pleased to announce SALOME version 3.2.3pre1. It is a preliminary version of maintenance release that contains the results of planned major and minor improvements and bug fixes against SALOME 3.2.2 version released in September 2006. This pre version is a candidate release for 3.2.3 official maintenance version.

### New features and general changes

#### Change of prerequisites

Prerequisites of Version 3.2.3pre1 have been changed.

The first change is Open CASCADE Technology 6.1.2a version. This is an alpha version which contains some fixes, especially connected with working on Mandriva 64 bit.

The second change is NETGEN. During the improvement of NETGEN performance for building 1-2D mesh, special changes were done inside NETGEN product. As usual these changes are packed as a patch inside NETGENPLUGIN\_SRC

#### Mandriva 64 bit qualification

OCC has carried out a test campaign on Mandriva 2006 64 bit and made special debug for running SALOME on this 64 bit OS. A lot of problems were fixed and the version now runs without serious problems. There are still some limitations (see the last section) which will be fixed by passing onto OCCT 6.1.2 final version (some patches on OCCT are under testing now were not included into this pre-release)

#### New partition algorithm

OCC has conducted a deep redesign of the partition algorithm. During this redesign the quality and performance of partition were significantly improved. API on new partition has also been changed. Only two parameters are necessary now: the object which is processed and the tool by which the algo makes section. Please refer to documentation.

*Important note:* due to the fact that this is a first version, there are possible limitations which are in progress of qualifications now (please refer to the last chapter). Possible limitations found by OCC or CEA/EDF will be corrected in the version 3.2.4.

#### smesh.py interface

smesh.py public python API has been improved and now provides the same level of functionality like the IDL interface before.

During this task all SMESH\_SWIG scripts have been rewritten for the new smesh.py interface. Documentation examples were accordingly updated.

**NETGEN 1-2D performance improvement**

Special improvement of NETGEN 1-2D mesher has been performed. The mesher is now ~25 times faster for b-spline surfaces and ~twice faster for analytical ones. For example, computation of 2D mesh on "flight.brep" model takes half a minute instead of 15 minutes as before.

**SALOME Install Wizard modifications**

Install Wizard has been modified to take into account the requirement to have a possibility to customize the buttons on the last "Finish Installation" page of the Installation Wizard.

OCC has implemented the support of an additional section for XML files (named <button>) which allows to add buttons and attach some script to these buttons. The script is responsible for the performance of certain actions (in addition this script is used to set the "enabled" state of the button - when no action can be done this script should give a signal about it).

Moreover, the existing "Launch SALOME" button is now also implemented with this feature. It means that this button can be completely removed from the Installation Wizard and replaced by another one (attribute <disable> of <button> section) .

**Possibility to export preferences from Python modules into standard "Preferences" widget of the SALOME Desktop has been added**

The possibility to export preferences from the Python modules to the common "Preferences" dialog box is available in SALOME.

To use it, implement in you MODULEGUI.py module method createSettings() and use SalomePyQt library to define the settings your module requires.

For example:

```

from qt import *
# The method createPreferences() is called automatically by SALOME GUI
# when the "Preferences" dialog box is invoked by the user.
# This method is called only once!
def createPreferences():
    # Import SalomePyQt module
    import SalomePyQt
    sg = SalomePyQt.SalomePyQt()
    # Create tab "Global" in the "Preferences" dialog box for your module's
page
    tid = sg.addPreference( "Global" )
    # Create a group of controls "General" on the page "Global"
    gid = sg.addPreference( "General", tid )
    # Let the group box "General" have two columns
    sgPyQt.setPreferenceProperty( gid, "columns", QVariant(2) )
    # Create "Display mode" combo box in the "General" group box.
    # Note that the name of the preference in the preferences file
    # is defined by the two last parameters.
    # You can always access this preference value by calling
    # sg.stringSetting("MODULE", "display_mode")
    # and
    # sg.addSetting("MODULE", "display_mode", "Shading")
    dispmode = sgPyQt.addPreference( "Display mode", gid,
                                     SalomePyQt.PT_Selector,
                                     "MODULE", "display_mode" )

    # Fill in the combo box with the values
    sl = QStringList(); sl.append("Wireframe"); sl.append("Shading")
    sgPyQt.setPreferenceProperty( dispmode, "strings", QVariant(sl) )

```

```

# Add the font setting to the "General" group box
sgPyQt.addPreference( "Font", gid,
                      SalomePyQt.PT_Font, "MODULE", "font" )
# Add a new "Other" tab
tid = sgPyQt.addPreference( "Other" )
# Create group "Print settings" in the "Other" tab
gid = sgPyQt.addPreference( "Sample", tid )
# ...
# etc

```

In general, preferences are added by the `addPreference()` method and different settings can be property set via the `setPreferenceProperty()` method.

SALOME GUI provides different controls which can be used in the "Preferences" dialog box: check box for boolean values; color, font and file selector tools; combo box for the list of possible choices; spin boxes for the integer and floating point values and other). Refer to the `QtListResourceEdit.*` files to learn about available types of control and their properties (for example, spin box for the floating point value has "min", "max", "step", "suffix", "prefix" and "special" properties).

Please, note that since all these methods accept `QVariant` type as a value parameter, you need to construct it explicitly (as in the example above).

Unfortunately currently there is no support of `QValueList<QVariant>` values in the PyQt toolkit. Thus if the user needs to set the preference property which takes the list of values (except the list of strings - in this case you can use `QVariant(QStringList())`), you can use `addPreferenceProperty()` continuously instead of `setPreferenceProperty()`.

For example, if you need to set custom values for the display mode items in the example above you can do the following:

```

# create string list
sl = QStringList()
# add first value
sl.append("Wireframe")
# add another value
sl.append("Shading")
# set strings to the combo box
sgPyQt.setPreferenceProperty( dispmode, "strings", QVariant(sl) )
# set value 10 for the first item (this value will be stored in the
# preferences file
sgPyQt.addPreferenceProperty( dispmode, "indexes", 0, QVariant(10) )
# set value 20 for the second item
sgPyQt.addPreferenceProperty( dispmode, "indexes", 1, QVariant(20) )

```

### Significant improvement of SALOME PyQT interface

SALOME PyQT interface has been improved to provide more functionality for python modules. See below what has been done:

1. Access to the log message output window from the Python modules has been implemented. The log messages output window is automatically shown by default when any Python module is activated. To disable this dockable window for some Python module, its `window(...)` callback function should be implemented in the Python GUI module.

The API of the `SalomePyQt` module has been extended by the following methods:

*Print message to the log messages output window:*

`message( <msg>, <add_sep> )`

where

`<msg>` is the message to be printed in the embedded log output window;

`<add_sep>` is a boolean flag: if it is `True` (by default) then the separator is

*printed in the log output window immediately after the message itself.*

*Clear the contents of the log messages output window:  
clearMessages()*

2. The API of the libSalomePy Python interface module has changed. All methods exported by this module: getRenderer(), getRenderWindow() and getRenderWindowInteractor() now accept boolean parameter "toCreate":
  - a. - if this parameter is not 0, the new VTK window is always created;
  - b. - if this parameter is 0 [default], a new VTK window is created only if no windows have been created by that moment; otherwise the existing window is activated and the corresponding VTK object is returned.
3. A new method has been implemented in libSalomePy Python module to show/hide the trihedron in the active VTK window: showTrihedron( <show> ) where <show> is a boolean flag; if VTK is not opened there, nothing happens.
4. Method UpdateView() has been implemented in the libSALOME\_Swig Python interface which redraws the contents of the current 2d/3d view:
 

```

import libSALOME_Swig
sg = libSALOME_Swig.SALOMEGUI_Swig()
sg.UpdateView()

```
5. Fixed bug: if custom VTK actors are displayed in the VTK viewer (e.g. via direct access to VTK renderer from the Python modules) which are not based on SALOME\_Actor - the viewer operations like "FitAll" work incorrectly.
6. libSalomePy module : new methods have been implemented to operate with the VTK viewer
  - a. fitAll() - to fit all contents of the current VTK viewer
  - b. setView(<type>) - to set the Top, Bottom, Front, Back, Left or Right view position
  - c. resetView() - to reset the view to the default state

### Successive animation mode

New animation mode is available now in VISU. If the user has different result files which contain computations of the same physical problem, and each file contains only a set of time stamps, now it is possible to animate all of them in one successive animation. Parallel animation works as before.

### Visu sweep animation improvement

Currently this improvement has been realized with two additional list-boxes in Post-Pro preferences of sweep:

- 1) Mode of sweeping – Linear (default)
  - Co sinusoidal
- 2) Parameter varies – From 0 to PI
  - From -PI to PI

### Visu – new kind of representation mode is available

"Surfaceframe" representation is now available for field data presentations (scalar map etc.). Similarly to the mesh presentation, this mode allows the user to see mesh edges on a shaded data presentation.

### Visu preferences look-and-feel improvement

During the improvement of animation and some other improvement, preferences of VISU module were also improved. Now they have a more logical view, all preferences are logically organized into tabbed sections.

**Visu – animation of bigger data is available now**

New option of the animation dialog box allows cleaning of memory at each frame of animation. This allows animation of greater amount of data with the same memory limit as before.

**Visu preferences have new option**

New option "Display only on creation" has been added to preferences to tab "Representation". It is intended to erase any existing presentations in a view except the newly created objects, their scalar bar and meshes.

**New mode during the creation of cut lines representation in Visu module**

Possibility to switch between relative (0...1) and absolute cut line length has been added. Default value of this option is controlled through Post-Pro module preferences.

**New parameter of "runSalome" script**

New command line parameter has been added for the runSalome.py script:

```
--test=[<hdf_file>|<python_script>][,<python_script>[,...]]
```

This parameter allows specifying the HDF file which should be automatically opened on the GUI desktop starting. In addition the user can specify one or more Python scripts to be imported in the opened study.

The files can appear in the arbitrary order. Only one HDF file can be specified.

If the HDF is not specified but Python script(s) appear with the `-test` parameter, a new empty study is created as alternative.

The Python scripts are imported in the order of their appearance in the option.

Caveats:

- a. No checks for the Python scripts are made. The user should ensure that his Python scripts are available (PYTHONPATH environment variable is set correctly). If any error takes place during the Python script importing, the rest of the scripts are ignored.
- b. If the extension is not given for some file - the ".py" is implied.

This option is available only in the GUI mode.

To import Python script(s) in the batch mode, use `--terminal (-t)` option.

**Rotation in Geom – new parameter**

New mode has been added to Rotation functionality. In this mode, the object can be rotated and three points are given as arguments (the central one and two points to compute the rotation axis and rotation angle). This new mode is available in both Rotation dialog box and Geometry module TUI interface, `GEOM_ITransformOperations::RotateThreePoints()` and `RotateThreePointsCopy()` IDL methods, corresponding IDL and `geompy.py` functions.

## Bug corrections

### Kernel

13553	<p><i>Summary:</i> bug in launchConfigureParser.py when GUI_ROOT_DIR is bad defined</p> <p><b>Changes :</b> launchConfigureParser.py script first checks if the GUI_ROOT_DIR environment variable is set properly and additionally checks availability of \${GUI_ROOT_DIR}/bin/salome/VERSION file.</p>
13554	<p><i>Summary:</i> -file option of runSalome</p> <p><b>Improvement :</b> New command line parameter has been added for the runSalome.py script: --test=[&lt;hdf_file&gt; &lt;python_script&gt;][,&lt;python_script&gt;[,...]]</p> <p>This parameter allows specifying the HDF file which should be automatically opened on the GUI desktop starting. In addition the user can specify one or more Python scripts to be imported in the opened study.</p> <p>The files can appear in the arbitrary order. Only one HDF file can be specified.</p> <p>If the HDF is not specified but Python script(s) appear with the --test parameter, the new empty study is created as alternative. The Python scripts are imported in the order of their appearance in the option.</p> <p><b>Caveats:</b></p> <ul style="list-style-type: none"> <li>- No checks for the Python scripts are made. The user should ensure that his Python scripts are available (PYTHONPATH environment variable is set correctly). If any error takes place during the Python script importing, the rest scripts are ignored.</li> <li>- If the extension is not given for some file - the ".py" is implied.</li> </ul> <p>This option is available only in GUI mode. To import Python script(s) in the batch mode, use --terminal (-t) option.</p>
13665	<p><i>Summary:</i> Problem with importing of dotted Python modules</p> <p><b>Changes :</b> The bug with impossibility to import dotted python modules (for example, xml.dom.minidom) has been fixed.</p>

### GUI module (IAPP)

10867	<p><i>Summary:</i> noexcepthandler option in GUI not working</p> <p><b>Improvement :</b></p> <p>Modifications have been introduced, allowing to fully disable signals catching. This is available only with OCCT version 6.1.1 or above.</p> <p>Signals catching now is regulated by the environment variable NOT_INTERCEPT_SIGNALS. If NOT_INTERCEPT_SIGNALS is not set or set to zero, signals will be converted into OCCT exceptions and will be caught by Salome.</p>
-------	---

	<p>If NOT_INTERCEPT_SIGNALS is set to 1, signals will be not caught. With "noexcepthandler" set to "yes" (or "-r" command line option) the NOT_INTERCEPT_SIGNALS environment variable will be set to 1 automatically.</p> <p><b>Changes :</b></p> <ol style="list-style-type: none"> <li>1. Environment variable NOT_INTERCEPT_SIGNALS is automatically set to 1 if "noexcepthandler" is set to "yes" or "-r" command line option is used.</li> <li>2. OSD::SetSignal() is not called if the environment variable NOT_INTERCEPT_SIGNALS is set to 1.</li> </ol>
10992	<p><i>Summary:</i> How to create preferences in python modules?</p> <p><b>New Features :</b> The possibility to export preferences from the Python modules to the common "Preferences" dialog box is available in SALOME.</p> <p>To use it, implement in you MODULEGUI.py module the method createSettings() and use SalomePyQt library to define the settings your module requires.</p> <p><i>For example:</i></p> <pre> ... from qt import * # The method createPreferences() is called automatically by SALOME GUI # when the "Preferences" dialog box is invoked by the user. # This method is called only once! def createPreferences():     # Import SalomePyQt module     import SalomePyQt     sg = SalomePyQt.SalomePyQt()     # Create tab "Global" in the "Preferences" dialog box for your module's page     tid = sg.addPreference( "Global" )     # Create a group of controls "General" on the page "Global"     gid = sg.addPreference( "General", tid )     # Let the group box "General" have two columns     sgPyQt.setPreferenceProperty( gid, "columns", QVariant(2) )     # Create a "Display mode" combo box in the "General" group box.     # Note that the name of the preference in the preferences file     # is defined by the two last parameters.     # You can always access this preference value by calling     # sg.stringSetting("MODULE", "display_mode")     # and     # sg.addSetting("MODULE", "display_mode", "Shading")     dispmode = sgPyQt.addPreference( "Display mode", gid,                                      SalomePyQt.PT_Selector,                                      "MODULE", "display_mode" )      # Fill in the combo box with the values     sl = QStringList(); sl.append("Wireframe"); sl.append("Shading")     sgPyQt.setPreferenceProperty( dispmode, "strings", QVariant(sl) )     # Add font setting to the "General" group box     sgPyQt.addPreference( "Font", gid,                           SalomePyQt.PT_Font, "MODULE", "font" )      # Add new "Other" tab     tid = sgPyQt.addPreference( "Other" )     # Create a group "Print settings" in the "Other" tab     gid = sgPyQt.addPreference( "Sample", tid )     # ... </pre>

	<pre># etc</pre> <p>In general, the preferences are added by the <code>addPreference()</code> method and different settings properties can be set via the <code>setPreferenceProperty()</code> method.</p> <p>SALOME GUI provides different controls which can be used in the "Preferences" dialog box:</p> <ul style="list-style-type: none"> <li>- check box for boolean values;</li> <li>- color, font and file selector tools;</li> <li>- combo box for the list of possible choices;</li> <li>- spin boxes for the integer and floating point values and other).</li> </ul> <p>Refer to the <code>QtListResourceEdit.*</code> files to learn about available types of control and their properties (for example, spin box for floating point value has "min", "max", "step", "suffix", "prefix" and "special" properties).</p> <p>Please, note that since all these methods accept <code>QVariant</code> type as a value parameter, you need to construct it explicitly (as in example above).</p> <p>Unfortunately currently there is no support of <code>QValueList&lt;QVariant&gt;</code> values in the PyQt toolkit. So if you need to set the preference property which takes the list of values (except the list of strings - in this case you can use <code>QVariant(QStringList())</code>), you can use <code>addPreferenceProperty()</code> continuously instead of <code>setPreferenceProperty()</code>. For example, if you need to set custom values for the display mode items in the example above you can do the following:</p> <pre>... # create string list sl = QStringList() # add first value sl.append("Wireframe") # add another value sl.append("Shading") # set strings to the combo box sgPyQt.setPreferenceProperty( dispmode, "strings", QVariant(sl) ) # set value 10 for the first item (this value will be stored in the # preferences file sgPyQt.addPreferenceProperty( dispmode, "indexes", 0, QVariant(10) ) # set value 20 for the second item sgPyQt.addPreferenceProperty( dispmode, "indexes", 1, QVariant(20) )</pre>
13420	<p><b>Summary:</b> EDF PAL 290 : Dump Study : the destruction of an object does not appear in the scripts</p> <p><b>Improvement :</b> "Delete" operation has been improved by an additional check: it is now forbidden to delete a geometrical object, if it acts as an input for creation of another geometrical object. All first-level children of Geometry component in a study are taken, and for each of them input objects (if any) are compared with a candidate for deletion. In case if coincidence is found, a warning message is shown, and the object is not deleted.</p>
13466	<p><b>Summary:</b> Wrong multiple selection processing in the OCC viewer</p> <p><b>Improvement :</b> Selection in OCC viewer has been correctly synchronized with Object Browser.</p>
13475	<p><b>Summary:</b> Wrong selection of objects after Display/Erase from Python console/module</p> <p><b>Improvement :</b> <code>GEOM_Swig::setColor()</code> method does not corrupt the selection now.</p>
13587	<p><b>Summary:</b> Improve "Active module" popup menu command</p>



	<p><b>Improvement :</b> When the user invokes a popup menu in the Object Browser on some item which belongs to the module which is not currently active, the popup menu provides command "Activate MODULE module" where "MODULE" is the name of the module to activate. In addition, if the user invokes the popup menu on the item which belong to the module which is not available in the current GUI session (e.g. this module was not listed in the --modules command line option) - the popup menu command "Activate module" does not appear at all.</p>
13671	<p><i>Summary:</i> Improvement of check_sip.m4</p> <p><b>Improvement :</b> Check for the sip tool has been improved: - support for the versions 4.3*, 4.4* is added - set SIPDIR variably directly if it is not set by the user</p>
13673	<p><i>Summary:</i> Improvement of check_pyqt.m4</p> <p><b>Improvement :</b> Check for the pyqt library has been improved: - try to find pyuic in PATH environment variable has been added - missing '\$' has been fixed - try to find qglobal.sip in \${PYQTDIR}/share/sip/qt has been added</p>

## GEOM module

11107	<p><i>Summary:</i> Make a rotation given an object, an axis and 2 points</p> <p><b>New feature :</b> Now a shape can be rotated given: - an object - an axis - 2 points in a perpendicular plan of the axis</p>
13394	<p><i>Summary:</i> EDF PAL 280 GEOM : disk building from circle</p> <p><b>New Features :</b> A possibility create faces from closed edges has been added.</p>
13412	<p><i>Summary:</i> EDF PAL 286 : GEOM Sketcher: Apply is selected if we click on Y</p> <p><b>Improvement :</b> - Focus is not moved on "Apply" button after preview redisplaying, so it stays where the user wants it. - Improved ring of focus widgets to simplify values entering without mouse (with Tab)</p>
13610	<p><i>Summary:</i> Deflection of a shape can't be changed from Python script</p> <p><b>Changes :</b> Method GEOM_Swig::setDeflection() in GeometryGUI_Swig.cxx is changed. This method corresponds to definition of a deviation coefficient of AIS_Shape (currently this is supported only in OCC Viewer). This value is used for computation of triangulation for surfaces presentation. Correction of bug consists in clearing of previous triangulation before definition of a new deviation value in order to force recomputation of triangulation.</p>

**MESH module**

13338	<p><i>Summary:</i> EDF PAL266 SMESH: no mesh update via Display Only after editing the hypothesis.</p> <p><b>Improvement :</b> Update feature has been included into the "Display" and "Display only" commands.</p>
12557	<p><i>Summary:</i> V3_2_0b2: SMESH documentation for scripts not up to date.</p> <p><b>Improvement :</b> New methods have been added to the Mesh class in smesh.py interface. Documentation updated with correct level of API.</p>
13364	<p><i>Summary:</i> EDF PAL271 : Med export with not enough space on disk : No Warning message</p> <p><b>New Feature :</b> Message box "Mesh export failed ..." appears now in case of MED export problems.</p>
13397	<p><i>Summary:</i> Upper case characters must be used for Smesh Pugin names !</p> <p><b>Improvement :</b> From now on both variants of environment variable XXX_ROOT_DIR (upper case and case sensitive) will be automatically available for NETGEN and GHS3D plugins. Before this change the user who defined NETGENPlugin_ROOT_DIR (case sensitive variant) in his Salome environment was forced to redefine NETGENPlugin resources in his .SalomeApprc.x.x.x file.</p>
13400	<p><i>Summary:</i> EDF PAL 281 : memory limitation and frozen desktop</p> <p><b>Improvement :</b> Now in case of a memory allocation problem the application returns error with message "Memory allocation problem".</p>
13458	<p><i>Summary:</i> Crash for hypothesis Number segments Distribution with analytic density</p> <p><b>Improvement :</b> The bug has been fixed by means of a patch to Open CASCADE version 6.1.2a integrated to this version of Salome.</p>
13549	<p><i>Summary:</i> The order of elements must be documented.</p> <p><b>Changes :</b> Added needed comments into file SMESH_Mesh.idl. Changes can be seen after generation of documentation.</p>
13570	<p><i>Summary:</i> Impossible to compute mesh, restored from study (with 2D algorithm, assigned on shell).</p> <p><b>Improvement :</b> Improved computation of mesh state for 2D algorithms, assigned on shells. Now 2D hypothesis is also applicable to shells.</p>
13571	<p><i>Summary:</i> "Create Mesh" dialog: selected geometry is changed after hypothesis creation.</p> <p><b>Improvement :</b> "Create Mesh" dialog improved: selected objects are protected from changes during</p>

	hypothesis creation.
13607	<p><i>Summary:</i> EDF PAL 313 : Random group colors when unassigned</p> <p><b>Improvement :</b></p> <ol style="list-style-type: none"> <li>1. No random attribute value now.</li> <li>2. If we don't check the "Color number" control the attribute will not be created.</li> </ol>
13626	<p><i>Summary:</i> GetShapeID() method works wrong</p> <p><b>Changes :</b></p> <p>New method GetShapeIDForElem() has been added. Existed method GetShapeID() has been changed.</p>
13637	<p><i>Summary:</i> EDF PAL 316 : SMESH Numbering Display nodes does not work with many groups of nodes</p> <p><b>Changes :</b></p> <p>Change implementation in method SMESHGUI::OnGUIEvent() for case 9010 in the existing switch().</p>

### MED module

13616	<p><i>Summary:</i> Portability gcc4.1.1</p> <p><b>Improvement :</b></p> <p>portability for gcc4.1.1 has been provided for MEDMEDM_Mesh</p>
7549	<p><i>Summary:</i> Salome tag V2_1_0 : segmentation fault after a write from medmem to a med file</p> <p><b>Improvement :</b></p> <p>Now segmentation fault does not occur in case of writing MEDMEM::FIELD that has not been read or filled</p>
12664	<p><i>Summary:</i> different field time value between Field API and med file</p> <p><b>Changes :</b></p> <p>The FIELD attribute _time is updated now in createFieldSupport and createFieldSupportPart1 methods with the value read from the MED-file function MEDpasdetempsInfo.</p>
13372	<p><i>Summary:</i> EDF PAL 63 : MED readFieldInFile method with an file which doesn't exist -&gt; COMM_FAILURE</p> <p><b>Changes :</b></p> <p>MED_SRC/src/MED/Med_Gen_i.cxx SALOME_MED::FIELD_ptr Med_Gen_i::readFieldInFile(...) method: Added "try/catch" mechanism in line where: "new MED(MED_DRIVER,fileName)"</p>
13659	<p><i>Summary:</i> A published field got from readFieldInFile() has no IOR in reopened study</p> <p><b>Improvement :</b></p> <p>In MED module, a field obtained from file using readFieldInFile() in python script and then added to a study, is now correctly restored when the study is saved and then reopened.</p>

**VISU module**

8166	<p><i>Summary:</i> [IMP] A flag "clean memory after each frame" in visu animation.</p> <p><b>Improvement :</b>  1. The following characteristics from the "Animation" dialog are available now from the Post-Pro preferences:  -- Speed  -- Cycled animation  -- Use proportional timing  -- Clean memory at each frame  2. The tab order in Post-Pro preferences has been changed: now it looks like as follows:  "MED import"- "Scalar Bar"- "CutLines"- "Sweep,Animation"- "Representation"-...</p> <p><b>Changes :</b>  Now we use the ProcessEvent mechanism from SALOME_Event instead of qApp-&gt;lock() and qApp-&gt;unlock().</p>
13174	<p><i>Summary:</i> EDF226 VISU : Sweep improvement</p> <p><b>Improvement :</b>  "Sweep" Post-Pro operation can now be performed in two modes: linear (the only mode previously supported) and cosinusoidal. In this new mode, displayed values vary with time according to the formula <math>F(t) = (1 - \cos(t))/2</math>. The current sweeping mode is selected by the user through Post-Pro module preferences, along with the time interval (from 0 to <math>\pi</math> or from <math>-\pi</math> to <math>\pi</math>).</p>
13374	<p><i>Summary:</i> EDF PAL 274 : VISU easily visualize element edges.</p> <p><b>Improvement :</b>  When we visualize a field, it is possible now to display the element edges easily. This is available from "Surfaceframe" pup-up</p>
13498	<p><i>Summary:</i> Wrong or missing MED profiles support implementation in VISU CORBA MED convertor</p> <p><b>New Features :</b>  MED_SRC/idl/MED.idl - Added new method, which returns the original sequence of numbers of profiles.</p> <p><b>Improvement :</b>  MED_SRC/idl/MED.idl  - Interface SUPPORT:  -- new method getNumberFromFile(...)</p> <p>MED_SRC/src/MEDMEM/MEDMEM_Support.hxx  - class SUPPORT:  -- new field _number_fromfile  -- new method "virtual inline MEDSKYLINEARRAY * getnumberFromFile()"  -- new method "inline const int * getNumberFromFile(MED_EN::medGeometryElement GeometricType)"  -- new method setpartial_fromfile(...)</p> <p>MED_SRC/src/MEDMEM_I/MEDMEM_Support_i.hxx  - class SUPPORT_  -- new method getNumberFromFile(...)</p> <p>VISU_SRC/src/VISU_I/VISU_CorbaMedConvertor.hxx</p>

	<ul style="list-style-type: none"> <li>- namespace VISU</li> <li>- - new struct TCSubProfile</li> <li>- - new struct TCGauss</li> <li>- - new struct TCGaussSubMesh</li> <li>- - new struct TCGaussMesh</li> </ul> <p><b>Changes :</b>  MED_SRC/idl/MED.idl  MED_SRC/src/MEDMEM/MEDMEM_Support.hxx  MED_SRC/src/MEDMEM/MEDMEM_Support.cxx  MED_SRC/src/MEDMEM/MEDMEM_MedFieldDriver22.hxx  MED_SRC/src/MEDMEM_I/MEDMEM_Support_i.hxx  MED_SRC/src/MEDMEM_I/MEDMEM_Support_i.cxx</p> <p>VISU_SRC/src/VISU_I/VISU_CorbaMedConvertor.hxx  VISU_SRC/src/VISU_I/VISU_CorbaMedConvertor.cxx</p>
13555	<p><i>Summary:</i> cutlines ; Option to see the length as abscissa</p> <p><b>Improvement :</b>  There is new Cutline option to see the real length of the line, instead of [0 1] interval.</p>
13556	<p><i>Summary:</i> display only option for new created objects and scalar bars.</p> <p><b>Improvement :</b>  A "Display Only" option for newly created objects and their scalar bar has been introduced. This option is available trough Preferences of VISU module</p>
13614	<p><i>Summary:</i> Improve VISU module persistence mechanism: do not use AttributeComment.</p> <p><b>Changes :</b>  A new attribute AttributeString (the name supposes a more general usage of this attribute) has been created. All mentions of AttributeComment in VISU_SRC have been replaced with AttributeString. Method CorrectSObjectType was extended to support loading of the documents with previous version of the VISU persistence format.</p>
13727	<p><i>Summary:</i> EDF 323 VISU : anim properties accessible via TUI</p> <p><b>New feature :</b>  The range of the animation scalar bar can be modified in a Python script now.</p>

## SUPERVISOR module

12630	<p><i>Summary:</i> Mandriva porting: crash on run dataflow in STUD-007 test scenario</p> <p><b>Improvement :</b>  Correction has been made. No crash now, scenario works fine.</p>
-------	--

## Study

13474	<p><i>Summary:</i> VISU-STUDY ? 3.2.2pre Error or HangUp or Crash</p> <p><b>Improvement :</b>  Some calls from VISU engine to GUI have been wrapped with SALOME_Event to prevent hang-ups and crashes.</p>
-------	--

**Other**

13543	<p><i>Summary:</i> SMESH / UNV import/export. Version of the UNV format</p> <p><b>Changes :</b> Added necessary comments about UNV version into file SMESH_Mesh.idl. Changes can be seen after generation of documentation. The user's documentation has been also updated.</p>
13669	<p><i>Summary:</i> Patch check_cppunit.m4 for 64 bits</p> <p><b>Improvement :</b> Check for cppunit tool has been improved: support of 64bit platforms has been added.</p>
13670	<p><i>Summary:</i> patch for netgen45ForSalome.patch for 64 bits on non-pentium</p> <p><b>Improvement :</b> An additional patch file has been provided for the NETGENPLUGIN mesh plugin module in order to support netgen 4.5 compilation on 64bit platforms.</p>
13857	<p><i>Summary:</i> No SHMessageStd in \${CASROOT}/src</p> <p><b>Changes :</b> sh configure file of the Installation Procedure for Open CASCADE has been updated. SHMessageStd resource has been replaced by SHMessage.</p>

** Processed and answered queries**

13077	<i>Summary:</i> EDF242 KERNEL/HXX2SALOME libtool.m4 build_configure problem with not "native" libtool
13551	<i>Summary:</i> EDF PAL 307 GEOM Fuse : Order of parameters

## Supported Linux distributions and pre-requisites

SALOME 3.2.3pre1 supports Mandrake 10.1, Debian Sarge, Mandriva 2006, RedHat 8.0, 9.0, RedHat Enterprise 4, Scientific Linux 4.2 and Mandriva 64 bit.

SALOME 3.2.3pre1 version has been mainly tested with the following pre-requisite list on Mandrake 10.1 platform.

Following prerequisites are valid for SALOME 3.2.3 version. Please note that we try to use as much as possible of native products.

	Mandriva 2006	Mandriva 2006-64	Debian Sarge	Mandrake 10.1	RedHat Enterprise 4	RedHat 8	RedHat 9	RedHat Scientific 4.2	RedHat Scientific 4.2
gcc	4.0.1	4.0.1	3.3.5	3.4.1	3.4.1	3.2	3.2	3.4.4	3.4.4
Tcltk	8.4.5	8.4.5	8.4.5	8.4.5	8.4.5	8.0	8.0	8.4.5	8.4.7
Python	2.4.1	2.4.1	2.3.5	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
Qt&msg2qm	3.3.4	3.3.4	3.3.4	3.3.3	3.3.3	3.3.3	3.3.3	3.3.3	3.3.3
Sip	4.2.1	4.2.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1
PyQt	3.14.1	3.14.1	3.13	3.13	3.13	3.13	3.13	3.13	3.13
Boost	1.32.0	1.32.0	1.32.0	1.31.0	1.31.0	1.31.0	1.31.0	1.31.0	1.31.0
Swig	1.3.24	1.3.24	1.3.24	1.3.24	1.3.24	1.3.24	1.3.24	1.3.24	1.3.24
OpenCASCADE Technology	6.1.2a	6.1.2a	6.1.2a	6.1.2a	6.1.2a	6.1.2a	6.1.2a	6.1.2a	6.1.2a
Qwt	4.2.0	4.2.0	4.2.0	4.2.0	4.2.0	0.4.1	0.4.1	4.2.0	4.2.0
OmniORB	4.0.7	4.0.7	4.0.7	4.0.5	4.0.5	4.0.5	4.0.5	4.0.7	4.0.7
Hdf	5-1.6.4	5-1.6.4	5-1.6.3	5-1.6.3	5-1.6.3	5-1.6.3	5-1.6.3	5-1.6.3	5-1.6.3
Med	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3
Vtk	4.2.6	4.2.6	4.2.6	4.2.6	4.2.6	4.2.2	4.2.2	4.2.6	4.2.6
Numeric	23.7	23.7	23.7	23.7	23.7	22.0	22.0	23.7	23.7
Graphviz	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	1.9	1.9	2.2.1	2.2.1
Doxygen	1.4.6	1.4.6	1.4.6	1.4.6	1.4.6	1.4.6	1.4.6	1.4.6	1.4.6
NETGEN	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
docutils	0.3.7	0.3.7	0.3.7	0.3.7	0.3.7	0.3.7	0.3.7	0.3.7	0.3.7

In addition, other versions of pre-requisites may also work.

## 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 script delivered with Installation Wizard. Call "build.csh -h" to see all parameters of this script.
  - *Important remark:* on RedHat 8 with the native automake-autoconf tools, sources of KERNEL from CVS can not be compiled. As a workaround there is specially prepared sources of KERNEL in Installation Wizard (after "make dist" step from Mandrake 10.1). They can be compiled with old prerequisites, but user must not call "build\_configure" step. To compile he must call "configure", "make", "make install" as usual. Because of this please don't use "build.csh" with "-b" option for KERNEL, because this option forces build\_configure step. Call of "build.csh -i -o" process compilation and installation on RedHat8 correctly.

## How to get the version and pre-requisites

The [SALOME 3.2.3pre1](#) pre-compiled binaries for Mandrake 10.1, Debian Sarge, Mandriva 2006, Mandriva64 and RedHat 8.0 can be retrieved from the PAL/SALOME FTP site (<ftp://www.opencascade.com>).

There are sources of modules inside, and user can build sources from scratch using “build.sh” script coming with installation procedure.

Alternatively, SALOME modules can be downloaded from the following CVS repositories:

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

**IMPORTANT!** Source files for version 3.2.3pre1 are available in CVS via tag **V3\_2\_3pre1**

There are two patches on **NETGEN** which are placed inside NETGENPLUGIN sources. One patch for all 32 bit platforms, other one is addition to first and should be applied only for Mandriva 64.

During the compilation on NETGEN from sources by Install Wizard, the patches are applied automatically to the standard NETGEN installation. You can download NETGEN 4.5 from CVS of their official site <http://www.hpfem.jku.at/netgen/>

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

- Netgen 1D-2D and 1D-2D-3D algorithm does not need definition of 2D and 1D algorithms and hypotheses both for mesh and sub-mesh. If you have defined 2D and 1D algorithms and hypotheses together with Netgen 1D-2D or 1D-2D-3D algorithm they will be ignored during calculation.
- SALOME supports reading of documents from the previous version, but documents created in the new version may not be opened in older ones.
- Due to a bug with changes of tolerance in OCT 6.1, GUI test scenario PROD 04 can not be played in TUI mode without a workaround. The workaround is to call the same Boolean operation cut twice
- During the compilation of OCT 6.1 by makefiles on a station with NVIDIA video card you may 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 crash (See /usr/X11R6/lib/libGLU.la). We suggest making links:
 

```
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
- In the current implementation of “Save VISU” state operation the parameters of Gauss view Partition mode are not stored. If a window has been partitioned and saved, it will be restored as non-partitioned. The same concerns the background color.
- Fails of display of some presentation on quadratic elements in VISU (cannot create animation for IsoLines, CutPlanes etc.) is inside of the VTK. Currently used version of the VTK library



(4.2.6) can not properly process the quadratic mesh elements (only ScalarMap and DeformedShape can be created only) that is presented in the MED file. Unfortunately it is impossible to replace or overload the VTK functionality outside of the library. This problem will be fixed automatically when we port the SALOME platform on the VTK 5.0 or higher version). This concerns Gauss viewer on quadratic elements. On some files with quadratic elements it is impossible to build gauss presentation.

- Problem of TestVisu20.xml failure in Supervisor still exists in version 3.2.3
- Step-by-step execution in SUPERVISOR on some graphs fails. This functionality is only a prototype and has not been finished completely
- Due to VTK 4.4 limitation, display of numbers of nodes in SMESH module has problems (some numbers disappear from the viewer)
- VTK presentation in GEOM was not completely finished and has problems with performance and memory usage. It desirable to use OCT viewer in GEOM module.
- MEFISTO algorithm fails on some cases
- MEFISTO algorithm sometimes produces different results on different platforms
- On some cases the number of triangles generated by MEFISTO may change from each attempt of building the mesh
- Due to some known opened bugs, some non regression TUI tests fail or have different results on different OS:
  - **Component:**
    - 13437: unstable work of "PrintPorts()" method  
/AddComponent/A1  
/TypesCheck/A0,A2,A3  
/united/A1
  - **Geometry:**
    - 13538: /boolean\_operations/C0 – the result of the operation “common” of two shells connected by faces is a compound containing two faces (instead of 1 face)
    - 13476: /boolean\_operations/C2,C7,D0,D5,E0 - section operation between two compounds produces a shape different from the one, produced by CASCADE
    - 13545: /boolean\_operations/D1,D6,E4 - cut operation between two compounds produces a shape different from the one, produced by CASCADE
    - 12200: /partition/F9 - partition algorithm produces an invalid shape
  - **Med:**
    - 13415: /SWIG\_scripts/M0 - hang up during the execution of "medcorba\_test.py" in TUI
    - 13414: /bugs/A0 - group of edges is lost after reading a med file via the MED component
    - 13323: "omniORB.CORBA.COMM\_FAILURE" is raised after trying to read "Mistrat\_import22.med" file  
/readStructFile/A4  
/readStructFileWithFieldType/A3  
Note: this problem is reproduced only on Mandriva2006 and not reproduced on Mandriva-64 version

- **Mesh:**
    - 12385: /mesh\_Quadratic/A6 - "RotationSweep()" of the quadratic face around its edge is not implemented yet
    - 13630: /elements\_edition/A6 - "ClearLog()" method is not implemented yet
    - 13269: /groups/A3-A5 - unstable work of the "StudyBuilder::LoadWith()"
    - 13078: 3D\_mesh\_NETGEN/A5,B5,B7 - mesh, based on the sphere is not computed via NETGEN  
Note: this problem is not reproduced on Mandriva-64 version but is reproduced on Mandriva 2006
    - OCC13657 (13624): 2D\_submesh/A3 - submesh, computed via NETGEN2D differs from the one, computed the same way on Mandrake, Mandriva, RedHat, Debian versions  
Note: this is reproduced only on Mandriva 64 bit
    - 13503: /mesh/bugs/A0,A1 - Triangle (Mefisto) produces a wrong number of the triangles (or doesn't produce)  
Note: this is reproduced only on Mandriva 64 bit
  - **Study:**
    - 13247: /std/copy\_paste/A0 - applying of paste of the object imported via the MED component leads to crash  
Note: This happens only on Mandriva 2006
  - **Supervisor:**
    - 12575: /spv/GNode/A0 - abort of the execution of the graph with Goto node
    - 13437: unstable behaviour of the "SuperV.CNode.PrintPorts()" method  
/spv/GNode/A0  
/spv/INode/A0  
/spv/LNode/A0  
/spv/SAMPLES\_python/A4
  - **Visu:**
    - 12538: /visu/Plot3D/F8,F9,G0 - SIGSEGV during creation of Plot3D presentation, because of VTK 4.2 limitation
    - 12688: /visu/united/B1 - incorrect work of the "View::ShowPart()" method
    - 13029: /visu/dump\_study/A6 - "Min", "Max" range values are not taken into account after setting them for the "Scalar Map On Deformed Shape" presentation
    - 13147: /visu/dump\_study/A7 - "StreamLines" are not restored correctly during the execution of the script generated by the "DumpStudy" functionality
    - 13246: /visu/SWIG\_scripts/A9,B1 - SIGSEGV appears in GUI during the execution of "visu\_view3d.py" and "batchmode\_visu\_view3.py" scripts
- End user documentation for Supervisor module was updated only in part of screenshots.
  - Results of some TUI non-regression testing are different on slow and fast computers. This is due to using in some test cases the functionality of GEOM from Supervisor in parallel nodes.

Due to the fact that Open CASCADE Technology (OCCT) does not thread safe in some geometry operations, in some hardware configurations there is parallel conflict access to some data and such supervisor graphs fail. This problem will be fixed in future. At this moment the workaround is usage of GEOM nodes subsequently.