Exporting ParaView Scenes to PDF3D

Exporting ParaView Scenes to 3D PDF

ParaView

PDF3D
29th May 2023
# Table of Contents

1 Introduction .......................................................................................................................... 4
1.1 About ParaView .................................................................................................................. 4
1.2 About PDF3D ..................................................................................................................... 4
1.3 Benefits .............................................................................................................................. 4
1.4 Problem Identified ............................................................................................................. 5
1.5 Solution Identified ............................................................................................................. 5
1.6 Features ............................................................................................................................. 5
1.7 Platforms .......................................................................................................................... 5
1.8 Installation .......................................................................................................................... 6

2 User Interface .......................................................................................................................... 9
2.1 Loading the PDF3 plug-in in ParaView ................................................................. 9
2.2 Save or export 3D scene in ParaView to 3D PDF ...................................................... 10
2.3 Rendering 3D PDF .......................................................................................................... 12
2.4 PDF3D Settings ............................................................................................................... 12
2.4.1 Exporting ..................................................................................................................... 12
2.4.2 Conversion ................................................................................................................... 14
2.4.3 Page Format ................................................................................................................ 15
2.4.4 3D Scene ..................................................................................................................... 16
2.4.5 3D Views ..................................................................................................................... 17
2.4.6 Annotations ............................................................................................................... 18
2.4.7 Sections ....................................................................................................................... 19
2.4.8 Controls ....................................................................................................................... 20
2.4.9 Buttons ......................................................................................................................... 21
2.4.10 Animations .................................................................................................................. 22
2.4.11 Security ....................................................................................................................... 23
2.4.12 JavaScript ................................................................................................................... 24
2.4.13 WebGL ......................................................................................................................... 24
2.5 Using VRPN from ParaView and 3D PDFs ................................................................. 25
2.6 Using ParaViewPlus with geo-referenced images (including GeoTIFF) ................... 26
2.7 Exporting custom camera views, custom buttons and JavaScript to 3D PDFs .......... 27

3 Scripting ................................................................................................................................ 29
3.1 Using PDF3D plug-in through Python scripts ......................................................... 29
3.2 Demonstrating Automation Using PDF3D plug-in script & pvpython .................... 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Specification of PDF3D pvpython parameters for PDF3DParaViewPlugin</td>
<td>31</td>
</tr>
<tr>
<td>4 Patches extensions</td>
<td>36</td>
</tr>
<tr>
<td>5 Plug-ins extensions</td>
<td>38</td>
</tr>
<tr>
<td>6 Working with Animations</td>
<td>42</td>
</tr>
</tbody>
</table>
Table of Figures
Figure 1.1: ParaView showing the PDF3D plug-in ................................................................. 7
Figure 1.2: ParaView showing the PDF3D plug-in settings ................................................... 7
Figure 1.3: ParaView PDF3D plug-in Options ........................................................................ 8
Figure 2.1: Opening the ParaView plug-in manager ................................................................. 9
Figure 2.2: Selecting PDF3DParaViewplug-in and Loading ................................................... 10
Figure 2.3: Setting auto loading option for PDF3DParaViewplug-in ....................................... 10
Figure 2.4: Begin to export 3D scene in ParaView to 3D PDF ................................................ 11
Figure 2.5: Save or export 3D scene in ParaView to 3D PDF .................................................. 11
Figure 2.6: Adobe Reader rendering a 3D PDF exported by PDF3D ParaView plug-in.............. 12
Figure 2.7: Various exporting settings to save or export 3D scene to 3D PDF ......................... 13
Figure 2.8: Various conversion settings to save or export 3D scene to 3D PDF .................... 15
Figure 2.9: Various page format settings to save or export 3D scene to 3D PDF ................... 16
Figure 2.10: Various 3D scene settings to save or export 3D scene to 3D PDF ..................... 17
Figure 2.11: Various 3D views settings to save or export 3D scene to 3D PDF ....................... 18
Figure 2.12: Various annotations settings to save or export 3D scene to 3D PDF ................. 19
Figure 2.13: Various sections settings to save or export 3D scene to 3D PDF ....................... 20
Figure 2.14: Various controls settings to save or export 3D scene to 3D PDF ....................... 21
Figure 2.15: Various button settings to save or export 3D scene to 3D PDF ......................... 22
Figure 2.16: Various animations settings to save or export 3D scene to 3D PDF .................. 23
Figure 2.17: Various security settings to save or export 3D scene to 3D PDF ....................... 24
Figure 2.18: JavaScript to export with 3D scene to 3D PDF ................................................. 24
Figure 2.19: Various parameters used when exporting 3D scene to WebGL using PDF3D engine ..... 25
Figure 2.20: Selecting GeoReferencedImageReader plugin to read geo-referenced image .... 27
Figure 3.1: Opening the Python shell in ParaView for PDF3D export ..................................... 29
Figure 3.2: Python script in ParaView for PDF3D export ....................................................... 29
Figure 3.3: PDF3D output generated through pvpython ....................................................... 30
Figure 6.1: PDF3D Settings Dialog with Animation Parameters ............................................. 42
1 Introduction

ParaViewPlus® is a product by PDF3D. PDF3D is a company by Apryse, leaders in document processing.

ParaViewPlus® is comprised of open-source software ParaView, extended with a collection of patches and plug-ins, developed by PDF3D and 3rd parties. These extensions resolve some limitations within the open-source version of ParaView and augment it by implementing new features and supporting new input and output file types. The central part of ParaViewPlus® is the PDF3D® ParaView plug-in.

PDF3D® ParaView plug enables ParaView to export scenes (typically with 3D content) to 3D PDF files, using the PDF3D technical report generation. The plug-in also supports exporting spreadsheets and 2D charts. The plug-in further enables users to export to 3D PDF through Python scripts for batch production.

1.1 About ParaView

ParaView is an application designed to aid the visualization and investigation of large scientific data. The design goals of ParaView include development of an open-source multi-platform visualization application that supports distributed computation models to process large data sets. It supports parallel visualization and rendering, 3D interactions, level of detailization, and scripting usage for the python language.

ParaView supports reading and writing a large number of scientific and technical data formats, with a wide range of data structures including structured and unstructured grids, regular grids, adaptive mesh refinement cells, polygons, points, lines. It supports time dependent data visualization and animation, with a suite of filters and menus to control the temporal data and animation. A comprehensive set of data analysis and visualization filters (based on the VTK library) include contouring, clipping, streamlines, streamribbons, glyph shapes, LIC, decimation, smoothing, and domain decomposition for parallel operations.

1.2 About PDF3D

PDF3D technical publishing technology software from Visual Technology Services Ltd (an Apryse company), provides a rich suite of data conversion, compression, document management and interfaces for generating rich 3D PDF technical reports. The PDF3D ParaView plug-in is a part of the PDF3D suite of products, and is supported by library services from the PDF3D-SDK Pro software development kit library.

1.3 Benefits

- Communicate your technical analysis, with tools suitable for Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD), Computation Electro-Magnetism (CEM), Geophysics, Engineering, Medical Imaging, Material Science, Numerical Simulation result post-processing.
- Save time, use immediately on the latest platforms without extra installs. plug-in is available as part of pre-compiled ParaView binary installations, loaded using the plug-in Manager menu.
- Adapt to your local needs, customization path available as safety-net. If you wish to integrate into your own ParaView built from source code, the PDF3D-SDK version of the plug-in is also available, using Qt, Cmake, Cpack, with C++ wrapper to PDF3D library.
- Share your results more effectively using 3D PDF format. Create highly compact, compressed, interactive representations of your ParaView visualizations, sometimes smaller than movie clip files.
• Show transient results, vary playback speed, and step through key events. Animation playback in PDF can be faster than within ParaView itself because no data pipeline is needed to update 3D view.
• Show your complete study in context with annotations and supporting data views. All main display window types are captured, including 3D, 2D, Charting, Spreadsheet, which may be appended to a report template.
• Simple to learn & use. Fully integrated plug-in controls inside ParaView client menu system.
• Add custom annotation – add title, caption, keywords or company logo to PDF page.

1.4 Problem Identified
The ParaView pipeline creates complex interactive visualizations, where users take time to adjust and setup specific investigations. Simple single-image screen-shots or AVI/MPEG movies do not capture the fidelity of visual results and do not allow for any viewer interaction. Screen-shots still need annotations for technical reports.

1.5 Solution Identified
The PDF3D ParaView plug-in is designed to capture full 3D visualization and animation into standard PDF documents. These documents can be viewed by anyone with the free Adobe Reader, enabling zoom, pan, rotate and close inspection of 3D result in detail. Full annotations and other supporting data are captured at the same time into the rich PDF report.

1.6 Features
• Controls integrated to main menus and tool bar
• Capture 3D View snapshot to 3D PDF, using simple file-save icon.
• Capture 2D, Spreadsheet table data to PDF.
• Append and Merge to create multi-page PDF reports.
• Animation View Integration, Capture Sequences of Data or Slices
• Highly compressed 3D Geometry Representation, using PRC-HCT encoding technology
• Flash Menu control toolbar in PDF for animation playback
• Full colormap texture representation capture
• Color Legend capture and Annotation Generated in PDF view margin
• Any number of logo, image, flash movie clips, sound may be added, make your own house-style
• Text blocks may be defined to create technical engineering reports.
• Multi-block, Multi-zone, polygon or grid data representations in the same view.
• PDF/Engineering ISO 24517-1:2008 Compliance option, with PDF XMP Metadata Keywords
• Strong AES-256bit security control PDF document encryption
• 3D Axes Cube Reproduced with floating 3D labels
• Each ParaView Object in view has local Visibility control within PDF.
• Automatic multiple view creation for PDF scene navigation
• PDF3D plug-in License only, no need for Acrobat or any other tool installation
• Python enabled, fully supports automated batch scripting through python attributes
• Fixed or Floating network license options
• User Guide and Developer Guide Documentation
• May be included in customized branded versions of ParaView.

1.7 Platforms
• ParaView 5.11
• Windows 10 64-bit
• Qt 5.15.2
• Python 3.9.13
1.8 Installation

The PDF3D ParaView plug-in, together with other extensions, comes as the part of ParaViewPlus release by Visual Technology Services Ltd. It is also possible to build PDF3D ParaView plug-in with local ParaView sources for advanced users as a part of PDF3D-SDK.

After installation, following environment variables should be set to make the plug-ins fully functional:

- **LANG** – This environment variable should be set only under Linux. It controls how fonts are used when generating the PDF file, it must be set so the document generator can place appropriate fonts in user's output. If the LANG requirements differs from what the user likes, then PDF3D tools from a sub-shell with a different set of settings should be run. PDF3D uses 'fontconfig' for font selections on Linux. If Arial is available (which is of course not always the case on Linux systems) it should always use Arial. If it is not available, fontconfig will try to find the best match or guess a font that could be used instead of Arial. It highly depends on the available fonts on Your system and the fontconfig configuration. PDF3D does NOT use /etc/sysconfig/i18n directly, please use the LANG environment variable instead. Example of setting LANG is “LANG=en_US”,

- **PDF3D_LICENSE_PATH** - The PDF3D ParaView plug-in has enabled "automatic server discovery", so if the license manager is running on the same machine, or another machine on the same local network, it should automatically connected. To explicitly control the network connection used by the TurboFloat license server, set the additional environment variable PDF3D_LICENSE_PATH. This should have a fully qualified IP address or DNS host name, followed by "%6300". Note that the "%" character is required here. The port number 6300 is used for the license server connection and may need to be open in your firewall settings. For running the server on the same system, you can use the loopback address, for example user can set variable PDF3D_LICENSE_PATH to 127.0.0.1%6300,

- **PATH** – It should be updated with $INSTALLATION_DIR/bin/ directory at the beginning,

- **LD_LIBRARY_PATH** – This environment variable should be set only under Linux. It should be updated with directories with shared library files (.so, .dylib). Under some systems where LD_LIBRARY_PATH is not used, /etc/ld.so.config should be updated by 'sudo ldconfig >absolute-path<',

- **PYTHONPATH** – It should be updated with directories with Python modules. Its usually subdirectory of installation directory which is parent of site-packages directory.

In case, after running paraview, you get black-square, edit /etc/environment file and add to it 'QT_GRAPHICSSYSTEM=native' line. This occurs rare and only on some Ubuntu (Linux) machines.

After installation and loading the plug-in, the PDF3D Plug-in enabled ParaView will look as shown in Figure 1.1. (The details of loading the plug-in (if not automatically loaded) are described in Section 2.1.)
The plug-in also provides various settings for the PDF3D export through the PDF3D Settings accessible from the File menu as shown in Figure 1.2 and Figure 1.3.
Figure 1.3: ParaView PDF3D plug-in Options
2 User Interface
The PDF3D plug-in (and other plug-ins which are part of ParaViewPlus) can be loaded anytime into ParaView through the ParaView plug-in manager if it is not auto-loaded to start with.

2.1 Loading the PDF3D plug-in in ParaView
1. Open the ParaView plug-in manager as shown in Figure 2.1, by selecting the Tools menu and Manage plug-ins.

![Figure 2.1: Opening the ParaView plug-in manager](image)
2. Select PDF3DParaView plug-in and press the Load Selected button as shown in Figure 2.2.

![Figure 2.2: Selecting PDF3DParaView plug-in and Loading](image)

3. Notice that the property column for PDF3DParaView plug-in changes to Loaded. If you need the PDF3D plug-in to auto-load next time you start ParaView, enable the Auto Load option in the plug-in manager. Continue to ParaView by selecting Close button. These steps are shown in Figure 2.3.

![Figure 2.3: Setting auto loading option for PDF3DParaView plug-in](image)

2.2 Save or export 3D scene in ParaView to 3D PDF

1. Having loaded the PDF3D plug-in and created something within ParaView, you are ready to export a 3D scene. Press the PDF3D Export button as shown in Figure 2.4. An alternative path is to use the Export option in the File menu and choose Files of Type to PDF Files (*.pdf).
2. Choose a new file name or select an existing file and press the Save button as shown in Figure 2.5.

---

*Figure 2.4: Begin to export 3D scene in ParaView to 3D PDF*

*Figure 2.5: Save or export 3D scene in ParaView to 3D PDF*
2.3 Rendering 3D PDF
The 3D PDF may now be rendered using Adobe Reader or Acrobat, as shown in Figure 2.6.

Figure 2.6: Adobe Reader rendering a 3D PDF exported by PDF3D ParaView plug-in

2.4 PDF3D Settings
The PDF3D ParaView plug-in provides various settings for saving or exporting the 3D scene from ParaView to 3D PDF as summarized in the following Figure 2.7 to Figure 2.19.

2.4.1 Exporting
Screenshot allows exporting ParaView scene as screenshot (static bitmap picture) or 3D scene (full 3D representation).

Magnify specifies the screenshot’s magnification factor creating images larger than normal display resolution.
Merge Mode sets whether new file will be created or existing should be updated with exported scene.

Page Number sets the page number on which scene will be exported, if the Merge Mode is set to Insert at Page.

Replace By sets how annotation which should be replaced will be selected (by it's hash value, number in entire document or number on specified page).

Annotation Number sets number of annotation, which will be replaced, in output file which already contains 3D annotation.

Hash Value sets hash value of annotation, which will be replaced, in output file which already contains 3D annotation.

Load Template sets template filename which will be used for replacing already existing 3D scene.

PDF/E Compliance sets the PDF/E compliance of the exported document. If set additional metadata is included in the PDF file, and fonts may be embedded.

On Existing Page toggles whether the page is updated or not, if the Merge Mode is set to Insert at Page.

Show Update Dialog shows the Update Dialog, if the output file already exists.

Open PDF is a toggle to open (show) the PDF file after export is completed. This option requires the Adobe Reader or Acrobat to be installed.

Export Binary PLY is a toggle which controls if exported PLY (one of the options) is binary or ASCII.

Apply PDF3D Colors To Texture Conversion sets if PDF3D mechanism for converting per-cell colors to texture will be executed.

Export Entire Pipeline allows exporting all items from ParaView pipeline with preserved node visibility.

![Figure 2.7: Various exporting settings to save or export 3D scene to 3D PDF](image)
2.4.2 Conversion

Compression (earlier Output Format) sets used 3D compression format. It can be chosen to be one of U3D, U3D-RHC, PRC or PRC-HCT. U3D is used for Adobe version 8 compatibility. RHC is the improved Right Hemisphere Compression extension, sensitive to the Quality factor. PRC-HCT is generally the highest level of file compression, sensitive to the spatial Tolerance factor. Pure PRC may be used for interchange with other 3D software.

Quality sets the U3D-RHC compression quality.

Tolerance Type controls if specified Tolerance is absolute or relative value. It could be Relative or Absolute for scenes exported to PRC-HCT compression format.

Tolerance sets the PRC-HCT compression factor. It affects the level of compression, and may cause model distortion at some levels (i.e. when too big values are specified).

Log Filename sets the file to which logging (progress / error) information will be written to.

Zoom Factor controls the exported 3D scene zoom factor.

Verbose Log toggles detailed logging flag.

Print Statistics sets the printing of statistics of exported 3D scene to log file flag.

Export MultiView allows export of all views from ParaView’s multi-view or only the current view. Other views may be some other 3D scene, but 2D plots or Spreadsheet views too, animated or not.

Export All Views To Single PDF Page enables exporting all ParaView views on single PDF page.

Styled Spreadsheet allows export of stylized spreadsheet from ParaView’s spreadsheet view.

Perform Scene Normalization allows scene normalization, i.e. translating and scaling of exported 3D scene to avoid jittering in output 3D PDF scene.

Enable Simplification turns ON the process of reducing total triangle count whenever possible preserving features of the model. Simplified models may increase interactive performance speed and reduce the storage requirements for the output file.

Only Subset Placement enabled, during simplification process, selects each simplified model vertex only from the existing vertex positions. The Default method (when OFF) computes optimal new vertex position, with less distortion error.

UV Weight is a specific weight factor of significance applied for Texture UV parameters in multi-dimensional simplification space. Weight 1.0 means that UV error vector with magnitude 1.0 equivalent to XYZ error vector with magnitude of scene diagonal. Note, this is a measure based on the texture coordinate mapping, not the actual texture image color. This measure is applied only when Texture UV coordinates are present on the model.

Boundary Weight is a weight factor specifically for boundary faces (face with only one adjacent simplex) in the simplification error measure. The high default value has the normally desirable effect of preserving details at boundary faces where possible. When simplices is triangle – face is edge, when line segment – face is end-point.

RGB Weight is a specific weight factor of significance for Color RGB values in multi-dimensional simplification space. Weight 1.0 means that RGB error vector with magnitude 1.0 equivalent to XYZ
error vector with magnitude of scene diagonal. This measure is used only where vertex RGB color values are present on the model.

*Point Weight* is a specific weight factor for point positions in the simplification error measure. For well-conditioned models the simplification process is driven by model planes formed by its triangles (or by model lines formed by its line segments) and such small default value for position weight does not affect overall process much. However, for degenerate models, like ones with all points lie on the same plane or line, even small contribution from point positions improves overall robustness. This weight is not used during simplification of point sets.

*Threshold Triangles Count* specifies approximate triangle count of the simplified model. For single-model scenes, this is an exact threshold for output triangle count. For multi-model scenes, it is an approximate threshold. Per-model *Threshold Triangles Count* is calculated basing on this parameter, average triangles per model diagonal and total triangles count.

*Threshold Lines Count* specifies the approximate line count of the simplified model. For single-model scenes, this is an exact threshold for output line count. For multi-model scenes, it is an approximate threshold. Per-model *Threshold Lines Count* is calculated basing on this parameter, average points per model diagonal and total point count.

*Threshold Points Count* specifies the approximate point count of the simplified model. For single-model scenes, this is an exact threshold for output point count. For multi-model scenes, it is an approximate threshold. Per-model *Threshold Points Count* is calculated basing on this parameter, average points per model diagonal and total point count.

![Figure 2.8: Various conversion settings to save or export 3D scene to 3D PDF](image)

2.4.3 *Page Format*

*Left, Right, Top and Bottom Margins* sets area’s margins inside of which the scene will be rendered.

*Width* and *Height* controls the width and height respectively of the exported page.

*Format* allows to choose a standard page width and height, such as *A4, Letter, … etc.* or *Custom.*

*Orientation* allows to choose between *Portrait* and *Landscape* page orientation.

*Page Units* sets in which page units are defined coordinates.
Border Line sets a border for the page and the corresponding Left, Right, Top and Bottom Margins values. Values are specified from the edge of the page.

Figure 2.10: Various 3D scene settings to save or export 3D scene to 3D PDF.

2.4.4 3D Scene

Use ParaView Background activates the use of ParaView’s background color.

Background, Ambient, Diffuse and Specular allows to choose corresponding new colors.

Use Original Normals toggles between using the original normals or allowing 3D PDF readers such as the Adobe Reader to compute them on the fly.

Show Toolbar controls if toolbar will be visible on exported scene.

Opacity sets the 3D scene’s opacity. Resulting opacity will be multiplication of Opacity and scene’s opacity, defined in ParaView.

Figure 2.9: Various page format settings to save or export 3D scene to 3D PDF
Navigation constrains the 3D view mouse to disable all or a part of possible interactions. It could be set to Unset, Allow, No Rotate, No Pan & Zoom or Disable.

Crease Angle sets crease angle which will be applied to all models in the scene. Specified value is in degrees (possible values are from range 0.0 – 180.0).

Render Mode sets default rendering mode of exported 3D scene. It could be set in many ways such as Solid, Transparent, Wireframe, Vertices, ... etc.

VCT Mode sets the various vertex colour texture options. Generally not needed for ParaView output.

Default View could be set to CAD, Geospatial, Anatomical or None. A North Compass icon is added to the scene in Geospatial mode. In None mode, only VTK View is generated.

Lighting Scheme sets default lighting scheme. It allows a varied choice such as Headlamp, White, Day, Night, CAD, ... etc.

Colorbar Type could be Gradient or Discrete.

ZFightingPrevention controls turning on/off mechanism against Z-fighting.

ZShiftScale controls scale parameter of Z-fighting prevention mechanism.

XAxis, YAxis, ZAxis and Scalebar controls exporting of PDF3D axes around the scene.

Poster Image sets image to be shown when 3D PDF is opened instead of 3D PDF scene. 3D PDF scene is activated on mouse click.

2.4.5 3D Views

Default View sets default viewing scheme. User can select one of already defined sets of views or define its-own set of views.

View Name Name of view to be added to view's list-box, in Adobe Reader 3D annotation's toolbar, in case of creating new set of views.
Camera Based Direction X, Y, Z coordinates of camera's based direction of view to be added.

Camera Vector Up X, Y, Z coordinates of vector up camera's direction of view to be added.

Add View, Remove View, Edit View and Save View adds new view with properties defined in user interface widgets to the list of views to be exported, remove currently selected view from the list of views to be exported, edits properties of currently selected view in the list of views to be selected, saves view properties to currently selected view in the list of views to be exported.

![ParaViewPlus GUI](image)

Figure 2.11: Various 3D views settings to save or export 3D scene to 3D PDF

2.4.6 Annotations
An arbitrary list of text, image or animation elements can be specified to annotate the PDF export page.

Left and Bottom specifies the corresponding relative margins of rectangle where annotation will be displayed.

Font Size specifies the text annotation’s font size.

Width and Height specifies the rectangle's width and height, inside of which the annotation will be displayed.

Rotation specifies the orientation angle of the displayed annotation.

Horizontal allows to choose between Left, Center and Right alignments. It sets how an annotation will be aligned with rectangle's vertical edges.

Vertical allows to choose between Top, Center and Bottom alignments. It sets how an annotation will be aligned with rectangle's horizontal edges.

Subplot allows only Page option currently. If sections are defined using the Section tab, these appear here. In case some other subplot is selected, the annotation's coordinate values are relative to subplot's coordinates.

Label sets the text annotation's label.
Media File sets the file name that is to be added as an annotation. Media may be a typical Image file (.png, .jpg, .tif), or a Flash (.swf) animation.

Media button allows to choose a Media File.

Add, Remove, Edit, Save and Load performs corresponding actions for annotations.

Figure 2.12: Various annotations settings to save or export 3D scene to 3D PDF

2.4.7 Sections
Annotiations may be organized into sections, where areas of the page may be dedicated to footers etc.

Left and Bottom sets the corresponding relative margins of the section displayed.

Width and Height sets the corresponding relative sizes of the section displayed.

Name sets the section name.

Docking allows rules such as Detached, Left, Right, Top, Bottom or No Frame. It sets if section will be docked to one of page's edges or detached.

Add, Remove, Edit and Save performs respective actions to the sections in the list.
2.4.8 Controls

Animation Control controls visibility of animation control in case animated 3D PDF is exported.

Use Flash Controls controls if Flash controls or native Adobe controls will be used in exported 3D PDF.

Z Scale Control enables it in the output for interactive changes to Z scale when viewing the 3D PDF.

Default Z Scale specifies the default value in Z scale control.

Show Toolbar enables showing toolbar control in the exported scene.

Override Colorbar Position enables predefined colorbar coordinates. By default, colorbar is exported right from 3D scene with page's margins corrected, so colorbar can fit the scene's exporting rectangle. In case this flag is turned on, page's margins are not corrected (i.e. user must update them) and colorbar will be drawn at the specified rectangle (LeftMargin, BottomMargin, Width, Height).

Left Margin and Right Margin sets the position (margin) of lower-left corner of the colorbar.

Width and Height sets the corresponding sizes of the colorbar.

Keywords specifies the PDF keywords which will appear in the PDF file properties dialog.

Title specifies the output PDF 3D scene title in bold over the top of the 3D view.

Caption specifies the output PDF caption displayed as a paragraph box under the 3D view.

AltText Under 3D sets text which will be displayed instead of annotation in PDF readers which do not support 3D PDF.

Custom Watermark Properties turns on using custom watermark properties.

Watermark Font Size specifies font size used for drawing watermark text.

Watermark Opacity specifies opacity of drawn watermark over 3D view.

Watermark Rotation Angle specifies angle (in degrees) for which watermark text is rotated (CCW).
Watermark Position X/Y specifies X/Y coordinates for watermark text lower left position.

Figure 2.14: Various controls settings to save or export 3D scene to 3D PDF.

2.4.9 Buttons
Left is left margin coordinate in specified units at which new button will be added.
Bottom is bottom margin coordinate in specified units at which new button will be added.
Width is width of new button which will be added.
Height is height of new button which will be added.
Caption is text on new button which will be added.
Button Type is type of button which will be added.
Load JS From File performs loading JavaScript source code from file and populating appropriate widget with it which will be attached to the new button which will be added.
Add Button, Remove Button, Edit Button and Save Button perform adding of new button with currently defined parameters to the list of buttons to be exported, removing of currently selected button in the list of buttons to be exported, editing button properties of currently selected button from the list of buttons to be exported and saving currently button properties to currently selected button in the list of buttons to be exported.

![ParaViewPlus Interface](image)

*Figure 2.15: Various button settings to save or export 3D scene to 3D PDF*

2.4.10 Animations

Enable Animation Capture enables export of the entire animation or only the current 3D scene.

Export Point Cache (OBJ + PC2) enables exporting animated data sets to OBJ + PC2 files, instead to multiple OBJ files, in case exporting to OBJ is selected.

Time Unit Label sets the time's unit label to show in animation for each timestep.

Start Frame specifies the starting frame of the animation to be exported.

End Frame specifies the ending frame of the animation to be exported.

Step specifies the increment of frames of the animation to be exported.

Initially Playing enables initially playing animation.

Forward Direction controls direction in which animation is played by default.

Repeat By Default enables repeating animation after it is played out in way defined by next parameter.

Repeat By Default Mode controls mode in which animation is repeated (normally, ping-pong, etc.).

Delay Multiplier controls delay between two animation's frames. For faster animation, set smaller values.

Visibility Buttons parameters controls visibilities of various buttons on animation control.

Care should be exercised while animating to avoid crashes and large files such as minimizing the scene geometry, and selecting a subset of time steps is advised.
2.4.11 Security

Security Parameters enables securing of the output document.

Owner Password specifies the document’s owner password for modifications, editing.

User Password specifies the document’s user password to initially open the document for viewing.

Allow Printing enables printing the document.

Allow Modifying enables modifying the document besides annotations, form fields or changing pages.

Allow Extraction enables text and graphic extraction.

Add or Modify allows text annotations or form fields.

Auto Fill In allows filling in the existing forms or signature fields.

Extract Data allows text and graphics extraction to support users with disabilities.

Assemble Document allows document assembly operations such as insert, create, rotate, delete pages or add bookmarks.

Print High-Resolution allows printing a high-resolution version of the document.

Encryption allows to choose a specific encryption algorithm among RC4 v1 (40bit), RC4 v2 (128bit), AES v2 (128bit) and AES v3 (256bit). AES-256 is recommended for strong security.

All the above PDF3D settings could be saved for future use or reset to defaults using the corresponding Save Settings or Reset Settings buttons. Also the License button is available to start the PDF3D license utility.
2.4.12 JavaScript

Document Level JavaScript defines document-level JavaScript code to be exported to 3D PDF.

Annotation Level JavaScript defines annotation-level JavaScript code to be exported to 3D PDF.

Load From File loads Document Level Java Script or Annotation Level Java Script from file and populates appropriate widget on user interface.

2.4.13 WebGL

Light Power Intensity value of scene light that equals physical light intensity of 1.0

Save Point Sets Controls if points will be saved in exported WebGL file(s)

Save Line Sets Controls if line segments will be saved in exported WebGL file(s)

Omit Defaults Controls if default values will be saved in exported WebGL file(s)
Add Asset ID Controls if asset identifier will be added as a prefix to exported WebGL file(s)

Use Alpha Blending Controls if alpha mode should be set to BLEND for meshes with transparency

Use Specular Glossiness Controls if specular / glossiness PBR model should be used. Enables specular color control.

Use Blinn Phong Controls if Blinn Phong model should be used

Convert sRGB to Linear RGB Controls if scene colors should be converted from sRGB to linear RGB colorspace before saving to WebGL

Use Draco Compression Controls if draco compression will be used when saving WebGL

Use Position Range Controls if explicit quantization will be used for vertex positions

Use Attribute Range Controls if explicit quantization will be used for vertex attributes

Point Threshold Maximal number of vertices in the model at which compression is not needed

Compression Level Sets the desired level of compression (0-10)

Position Bits Number of quantization bits for vertex position

Normal Bits Number of quantization bits for vertex normals

Texture Coordinate Bits Number of quantization bits for vertex texture coordinates

Color Bits Number of quantization bits for vertex colors

![ParaViewPlus Settings](image)

**Figure 2.19:** Various parameters used when exporting 3D scene to WebGL using PDF3D engine

### 2.5 Using VRPN from ParaView and 3D PDFs

If ParaView package contains VRplug-in plug-in, users can use Space Navigator and other 3D motion controllers supported through VRPN interface. To use these devices, following steps should be performed:

- In case 3D device drivers are not automatically started, start the drivers manually.
• Open vrpn.cfg (comes with ParaView or VRPN package, depending on the customer's type) in text editor and remove commenting of a following line, or another one, depending on device user has (caution, only one line should be active at the moment):

\[ vrpn_3DConnexion_Navigator device0 \]

• Start vrpn_server with passed vrpn.cfg file by executing from command-line:

\[ vrpn_server -f vrpn.cfg \]

• Create new VR Connection on VR Panel by clicking on Add button (in VR Connections section). Here is a brief description of connection's parameters:
  - **Name** – connection's name (can be some arbitrary string),
  - **Address** – address of device which will be used for 3D navigation. It's syntax is \( device\_name@machine\_uri \) (in case of above vrpn.cfg file – \( device0@localhost \)),
  - **Inputs** – which device's actions will be processed. Here is brief description of possibilities:
    - **Analog** – in case of Space Navigator, these inputs will transmit tracker's moves. Space Navigator has only one Analog input which ID is 0 (add it),
    - **Buttons** – in case of Space Navigator, right button has ID 0, while left button has ID 1 (add them both),

• Select the interaction parameters on VR Panel and add a new interaction by clicking on Add button (in Interactions section). Here is brief description of interaction's parameters:
  - **Interaction Type** – parameter in the right-most combo-box, specifies the type of interaction. Select Space Navigator Grab or some other type
  - **Property to alter** – parameter in the middle combo-box, specifies which parameter will be changed by Space Navigator's actions. In this case, select Model or Eye Transformation Matrix,
  - **Object which property will be altered** – parameter in the left-most combo-box, specifies the properties of which object will be altered. Select one of RenderView objects,

• Connect ParaView with VRPN server by clicking on Start button on VR Panel.

Users can also use Space Navigator in Adobe Reader to perform easier 3D navigation. After installing the device, open 3D PDF, select 3D scene and navigate through it using Space Navigator freely. There are no other steps required to be performed for its usage.

### 2.6 Using ParaViewPlus with geo-referenced images (including GeoTIFF)

ParaViewPlus can interpret various geospatial data, including grid file formats (ESRI ArcGIS, ZMAP, etc), but, also geo-referenced images, including GeoTIFFs.

Geo-referenced images support is provided, through GeoReferencedImageReader reader plugin. Using this reader plugin, geo-referenced TIF, TIFF, JPG, JPEG and PNG files may be loaded into ParaView and processed to be later exported as 3D PDF through the PDF3D ParaView plugin.

Please note that there is already some limited support for these file formats in base ParaView, if functionality seems limited, check that the GeoReferencedImageReader reader plugin is correctly loaded and has been selected as the correct input reader, as shown in Figure 2.20.
2.7 Exporting custom camera views, custom buttons and JavaScript to 3D PDFs

Along with the already existing views, users can export custom camera views, by using parameters on the 3D Views tab in PDF3D Settings dialog. There are already several predefined set of views (CAD, Geospatial, Anatomical, etc.), but, if user wants it’s own set of views, Custom should be selected (for Default View parameter) and new views, defined in View Name, Camera Base Direction and Camera Vector Up parameters, can be added by pushing Add View button. After adding new custom view, it's definition will be displayed in below list of custom views. Already existing set of views can be manipulated using Remove, Edit and Save View buttons by selecting particular custom view and pushing one of the buttons. Custom views can also be added from pvpython environment.

User can export custom buttons to 3D PDF by using parameters from Buttons tab in PDF3D Settings dialog. New button can be added by defining button margins Left and Bottom, format Width and Height, Caption, Button Type and JavaScript action executed when the button is pressed. After these parameters are defined, user can add button with these properties using Add Button button. JavaScript action code can be manually written or copy-pasted, but, also, can be loaded from file. After adding new custom button, it's definition will be displayed in below list of custom buttons. Already existing buttons can be manipulated using Remove, Edit and Save buttons. Custom buttons can also be added from pvpython environment, example which shows this functionality is export_buttons.py.

In some cases, we want to pre-define current 3D scene behavior (mouse behavior, buttons actions, etc). These actions are implemented using JavaScript. User can change document level and annotation level JavaScript by using parameters on JavaScript tab in PDF3D Settings dialog. For changing document level JavaScript, there is Document Level JavaScript parameter and for changing...
annotation-level JavaScript, there is *Annotation Level JavaScript* parameter. These parameters (JavaScript) can be set up manually or can be loaded from file. These parameters can also be used (like all other parameters in PDF3DParaViewPlugin) from pvpython environment also, example which shows this functionality is `export_js.py`. 
3 Scripting

The PDF3D ParaView plug-in also supports Python scripting through ParaView. PDF3D is also supported under pypython (no graphical user interface or from Python Shell), which is essential for batch processing and automation. All PDF3D Settings which are available through ParaView's File menu could be also set through Python scripts allowing for complete automation.

3.1 Using PDF3D plug-in through Python scripts

1. Python scripting of PDF3D export in ParaView could be demonstrated by selecting Python Shell from the Tools menu as shown in Figure 3.1.

   ![Figure 3.1: Opening the Python shell in ParaView for PDF3D export](image1)

2. Then running the below minimal Python script in the Python shell, as shown in Figure 3.2, will have the same effect that was explained in the earlier chapter of using PDF3D plug-in through the ParaView user interface.

   ```python
   exporters = servermanager.createModule("exporters")
   exporter = exporters.PDF3DExporter()
   exporter.FileName = "test.pdf"
   exporter.View = GetActiveView()
   exporter.Write()
   ```

   ![Figure 3.2: Python script in ParaView for PDF3D export](image2)
3.2 Demonstrating Automation Using PDF3D plug-in script & pvpython

The below script may be launched by calling ‘pvpython shock.py’ on command line. The 3D PDF output is generated, without any user intervention, and is shown in Figure 3.3.

The example shows how to load state from a particular ParaView state file, alter scene parameters, save the resulting scene to PDF3D file using the plug-in. In case of exporting animated scenes using pvpython, its mandatory to set exporter’s parameter $\text{AnimationName}$, which is the name with which the animation scene we want to export to PDF3D is registered with Proxy Manager.

```python
#!/usr/bin/pvpython
from paraview.simple import *

# Find PDF3D dynamic library
mydll_path = '<path here>/PDF3DParaViewplug-in.dll')

# Load PDF3D plug-in library
servermanager.Loadplug-in(mydll_path)

# Load state file
my_statefile = "shock.pvsm"
servermanager.LoadState(my_statefile)

# Gets default render view
view = servermanager.GetRenderView()
```

![Figure 3.3: PDF3D output generated through pvpython](image_url)
# Retrieves PDF3DExporter exporter object
exporters = servermanager.createModule("exporters")
exporter = exporters.PDF3DExporter()

# Sets exporter’s arguments
exporter.InputStateFile = my_statefile  # same as pvsm
exporter.FileName = ‘shock.pdf’
exporter.View = view
exporter.UseParaViewBackground = 1
exporter.PageOrientation = 1
exporter.BottomMargin = 85
exporter.BorderLineLeftMargin = 10
exporter.BorderLineRightMargin = 10
exporter.BorderLineTopMargin = 10
exporter.BorderLineBottomMargin = 10
#exporter.Title = ‘ParaView Shock Model’

# set full 3D or screenshot mode:
exporter.Screenshot = 0

# Exports current 3D scene
exporter.Write()

3.3 Specification of PDF3D pvpython parameters for PDF3DParaViewPlugin
In this section are described all PDF3D ParaView Plugin parameters which could be used from pvpython environment (complete list could be accessed by executing ‘dir(exporter)’):

- **FileName** – Output file name to which scene will be exported (.pdf, .prc, .u3d, .obj, .ply),
- **OutputFormat** – Compression format of exported 3D mesh (0-U3D, 1-RHC, 2-PRC, 3-HCT),
- **QualityFactor** – Compression quality factor for previously selected U3D OutputFormat,
- **ToleranceType** - Type how tolerance is calculated (false - absolute or true – relative),
- **ToleranceFactor** - Tolerance factor used when exporting to PDF3D file with one of PRC compressions types,
- **MergeMode** - In case output file exist, use merge mode (0 - new, no merge, 1 - prepend, 2 - append, 3 - insert, 4 - replace existing annotation),
- **PageNumber** - Insert scene on specified page number in output file if merge mode is set to insert,
- **TemplateMode** - Template mode in case template file is specified,
- **TemplatePageNumber** - Page number of template file at which is template which should be used,
- **Screenshot** - Exports scene as PDF3D screenshot annotation,
- **Magnification** - Annotation's magnification factor,
- **LogFilename** - Filename for logging during exporting process,
- **VerboseLog** - Sets logging process to be verbose,
- **PrintOnStdout** - Prints logging messages on standard output,
- **PrintStatistics** - Prints statistics after file is exported,
• **ZoomFactor** - Zoom factor of scene in output file,
• **PDFECompliance** - PDF/E Compliance of exported PDF document,
• **OnExistingPage** - Export on existing page instead of creating new page in output file (in case of merging),
• **OpenPDF** - Opens output PDF file after being exported,
• **ExportBinaryPLY** - Controls if exported PLY will be in BINARY or ASCII format,
• **ApplyPDF3DColorsToTextureConversion** - Controls if PDF3D conversion of scalars to texture is performed (for per-cell coloring),
• **StyledSpreadsheet** - Exports styledspread sheet (special colors, without header, used first line content for title, etc.),
• **EnableSimplification** - Flag controls if per-model simplification is enabled,
• **OnlySubsetPlacement** - Flag controls if simplification is performed by only subset placement,
• **ThresholdTrianglesCount** - Threshold triangles count used in simplification process,
• **ThresholdLineSegmentsCount** - Threshold line segments count used in simplification process,
• **ThresholdPointsCount** - Threshold points count used in simplification process,
• **BoundaryWeight** - Boundary weight value used in simplification process,
• **UVWeight** - UV weight value used in simplification process,
• **PointWeight** - Point weight value used in simplification process,
• **RGBWeight** - RGB weight value used in simplification process,
• **ReplacedAnnotationMode** - Mode of selecting which annotation will be replaced (0 - By hash value, 1 - By hash value at specified page, 2 - By numbered annotation on specified page),
• **ReplacedAnnotationNumber** - Which annotation on PageNumber will be replaced,
• **ReplacedAnnotationHashValue** - Annotation with specified hash value will be replaced,
• **TemplateFilename** - Template file which will be used for replacing 3D annotations,
• **LeftMargin** - Exported page left margin,
• **TopMargin** - Exported page top margin,
• **RightMargin** - Exported page right margin,
• **BottomMargin** - Exported page bottom margin,
• **PageWidth** - Exported page width,
• **PageHeight** - Exported page height,
• **PageFormat** - Predefined page format (A4, B2, etc.),
• **PageOrientation** - Exported page orientation,
• **PageUnit** - Page units,
• **BorderLine** - Flag controls showing of border lines,
• **BorderLineLeftMargin** - Exported page border line left margin,
• **BorderLineTopMargin** - Exported page border line top margin,
• **BorderLineRightMargin** - Exported page border line right margin,
• **BorderLineBottomMargin** - Exported page border line bottom margin,
• **XAxis** - Use custom PDF3D axes along x-axis,
• **YAxis** - Use custom PDF3D axes along y-axis,
• **ZAxis** - Use custom PDF3D axes along z-axis,
• **Scalebar** - Use custom PDF3D scalebar along selected axes,
• **UseParaViewBackground** - Use custom ParaView’s background color instead of using specified,
• **AmbientColor** - Default ambient material color for exported objects,
• **DiffuseColor** - Default diffuse material color for exported objects,
• **SpecularColor** - Default specular material color for exported objects,
• **BackgroundColor** - Default background color for exported scene,
• **BackgroundColor2** - Default second background color for exported scene (important for making gradient backgrounds),
• **ShowToolbar** - Show toolbar in exported PDF3D,
• **UseOriginalNormals** - Use original normals instead of computing them on-the-fly in Adobe Reader,
• **ZFightingPreventionEnable** - Enables Z-Fighting prevention mechanism,
• **ZFightingPreventionScale** - Specifies value for Z-Fighting Prevention scale. It is empirical coefficient. If there is Z-Fighting, then try to increase scale,
• **CreaseAngle** - Crease angle (in degrees) for exported PDF3D scene,
• **Opacity** - Opacity factor for exported PDF3D scene,
• **RenderMode** - Default render mode for exported scene,
• **LightingScheme** - Default lighting scheme for exported PDF3D file,
• **VCTMode** - Used VCT mode (Vertex Color Textures) instead of using per-vertex colors,
• **ColorbarType** - Colorbar’s type (independent boxes or gradient),
• **NavigationMode** - Navigation mode of output file,
• **PosterImageFile** - Poster image file to be used for displaying 3d annotation while it is disable,
• **DefaultView** - Default view in exported document (CAD or Geospatial),
• **ListOf3DViews** - List of custom 3D views which will be exported to 3D PDF,
• **AnnotationsFile** - Annotations file (.pcfg) used for plotting parameters,
• **AnnotationsList** - Annotations used to add text or multimedia annotations to exported PDF3D file,
• **SectionsList** - Sections used to add subplot capabilities to exported PDF3D file,
• **ShowAnimationControl** - Show animation control in exported PDF3D,
• **UseFlashControls** - Flag controls using of Flash PDF controls or non-Flash,
• **ShowZScaleControl** - Show Z scale control in exported PDF3D,
• **DefaultZScale** - Default Z scale of exported scene,
• **OverrideColorbarPosition** - Flag controls using of custom colorbar coordinates,
• **ColorbarOrientation** - Value controls colorbar orientation (0 (default) - vertical, 1 – horizontal),
• **ColorbarLeftMargin** - Colorbar’s left margin,
• **ColorbarBottomMargin** - Colorbar’s top margin,
• **ColorbarWidth** - Colorbar’s right margin,
• **ColorbarHeight** - Colorbar’s bottom margin,
• **Keywords** - Keywords for exported PDF3D file,
• **Title** - Output PDF3D page title,
• **Caption** - Output PDF3D page caption box,
• **AltTextUnder3D** - This area requires a 3D PDF enabled viewer such as Adobe Reader,
• **Watermark** - Watermark text for exported PDF3D file,
• **CustomWatermarkProperties** - Flag controls if custom watermark parameters should be used,
• **WatermarkFontSize** - Custom value for watermark font size,
• **WatermarkOpacity** - Custom value for watermark opacity,
• **WatermarkRotationAngle** - Custom value for watermark rotation angle,
• **WatermarkXPosition** - Custom value for X coordinate of watermark position,
• **WatermarkYPosition** - Custom value for Y coordinate of watermark position,
• **ButtonList** - List of buttons which will be exported to 3D PDF,
• **EnableAnimationCapture** - Take snapshot instead of exporting animation,
• **ExportPointCache** - Export OBJ + PC2 files instead of exporting sequence of OBJ files,
• **StartFrame** - Export animation from Start to End Frames,
• **EndFrame** - Export animation from Start to End Frames,
• **Increment** - When exporting animation export StartFrame + i * Increment until EndFrame is meet,
• **AnimationName** - Routine saves name of animation's scene which will be exported to PDF3D,
• **TimeUnitLabel** - Time's unit label for exported animated PDF3D file,
• **OwnerPassword** - Owner’s password for exported PDF3D file,
• **UserPassword** - User’s password for exported PDF3D file,
• **Permissions** - Permissions flags for exported PDF3D file,
• **Encryption** - Encryption parameter for exported PDF3D file (0 - RC4 v1 (40bit), 1 - RC4 v2 (128bit), 2 - AES v2 (128bit), 2 - AES v3 (256bit)),
• **DocumentLevelJavaScript** - Document's level JavaScript to be exported with PDF3D file,
• **AnnotationLevelJavaScript** - Annotation's level JavaScript to be exported with PDF3D file,
• **LightPower** - Intensity value of scene light that equals physical light intensity of 1.0
• **SavePointSets** - Specifies if point sets will be saved
• **SaveLineSets** - Specifies if line sets will be saved
• **OmitDefaults** - Specifies if set fields with default values should be saved
• **AddAssetID** - Specifies if asset id should added at the start of it's file name
• **UseAlphaBlending** - Specifies if alpha mode should be set to BLEND for meshes with transparency
• **UseSpecularGlossiness** - Specifies if specular/glossiness PBR model should be used. Enables specular color support
• **UseBlinnPhong** - Specifies if Blinn Phong model should be used
• **ConvertsRGBToLinearRGB** - Specifies whether to convert the scene's color values from sRGB to linear RGB before saving to glTF
• **UseDracoCompression** – Controls if draco compression will be used for saving WebGL files
• **UsePositionRange** - Specifies if explicit quantization will be used for vertex positions
• **UseAttributeRange** - Specifies if explicit quantization will be used for vertex attributes
• **PointThreshold** - Maximal number of vertices in the model at which compression is not needed
• **CompressionLevel** - Sets the desired level of compression
• **PositionBits** - Number of quantization bits for vertex position
• **NormalBits** - Number of quantization bits for vertex normals
• **TextureCoordinateBits** - Number of quantization bits for vertex texture coordinates
• **ColorBits** - Number of quantization bits for vertex colors
• **InputStateFile** - Routine loads PDF3D parameters from state file,
• **OutputStateFile** - Routine saves PDF3D parameters to state file previously populated with ParaView parameters,
• **View** - VTK View to be exported to PDF3D,
• **Write()** - Routine called to export currently active ParaView scene to PDF3D.
4 Patches extensions

The ParaViewPlus package is created by extending the ParaView framework with multiple patches and plug-ins. This section provides a brief description of patches added to ParaView (from v5.7.0) as a part of ParaViewPlus:

1. Modified ParaView splash-screen and resource images by adding PDF3D Logo to them,
   - The ParaView start-up splash screen is a normal png image file. We provided slightly modified version of it with PDF3D logo inserted. Same change was performed on the image shown in the Help-About dialog.

2. Added support for value dependent color-bar labels formats,
   - Sometimes, displaying more descriptive labels than numerical labels beside color-bar is required. Our patch allows adding color-bar text labels which depend on the value with which the labels will be mapped by extending syntax of the currently existing label format mechanism. If 'Automatic Label Format' is turned off, ParaView will, for every label, search for \{value1="string1" value2="string2" ...\} in label format and if label's value matches to one of valueN in {}, it will replace the entire construct with an appropriate string, the value is paired with.

3. Added support for environmental variables in state files, indicator about missing files, scrollbar area to "Fix Paths in State File" dialog and pipeline source name as a title for every proxy,
   - State files with filenames often require to be updated with locations of used files on a new machine. Our patch makes this process easier and faster by allowing pathnames to contain environment variables and by adding indicators to "Fix Paths in State File" dialog about a missing filename or directory. Also, scrolling area is added to the dialog to handle cases if there is a large number of input files specified in state file. In case of environment variables, supported syntax is %VAR_NAME%, ${VAR_NAME} and $VAR_NAME where VAR_NAME must be from [a-zA-Z0-9_]++ language.

4. Added interactive lighting support ParaView render-views,
   - Complex geological surfaces and terrains, are often best shown at certain oblique lighting angles, so the interactive discovery of best angle is needed. Our patch allows user to rotate currently used light sources, in the ParaView scene by using "Track light" option from "Camera Controls" toolbar, and perform their surface investigation with the optimally set lighting.

5. Added converting scalars to texture in VRML exporter before exporting of current scene to VRML
   - To get more natural output, our patch converts scalars to color texture which applies to input pipeline elements before they are exported to VRML file.

6. Added the patched version of main CMakeLists.txt to allow custom package name
• To be able to control application’s name and version strings, we introduced this patch. Now, we could provide more detailed information about product’s version and type we deliver.

7. **Added patch for allowing using Space Navigator from ParaView**
   • Notifications from Space Navigator come as analog notifications. Because of that, we were unable to use it with ParaView through its VRplug-in, because it expected tracker notifications. Our patch introduces a new Tracking type “Space Navigator Grab”, which allows users to use Space Navigator to be used in ParaView as a 3D navigation device.

8. **Added patch for naming solids by filename in ParaView’s STL writer**
   • Added extracting file name from output file path and setting solid name with it.

9. **Added shortcut buttons to splash-screen and Help menu**
   • For easier navigation, we added shortcut buttons to License Utility, User Guide, License File and PDF3D Web Site to application’s splash-screen and Help menu.

10. **Added extending list of standard readers with SEG-Y reader**
    • Added support for loading SEG-Y file format files. Both 2.5D and 3D are supported.
5 Plug-ins extensions

The ParaViewPlus package is created by extending ParaView framework with multiple patches and plug-ins. In this section are listed and briefly described plug-ins added to ParaView v4.0.1 as a part of ParaViewPlus:

1. **PDF3DParaViewplugin** – exports current scene to 3D PDF, PRC or U3D file formats,
   - *Description*: Exporter plug-in exports ParaView scenes (typically with 3D content) to 3D PDF files (.pdf, .prc, .u3d) using the PDF3D technical report generation. It also supports exporting spreadsheets and 2D charts as screenshots. The plug-in further enables users to export to 3D PDF through Python scripts for batch production.
   - *License*: Owned and developed by Visual Technology Services Ltd.

2. **3DDEM** – reads 3D uniform volume data, defined by ASCII data similar to CSV file,
   - *Description*: Reader plug-in allows reading 3D uniform volume data (.csv), defined by ASCII data similar to CSV file. The header contains some dimensional and geospatial data similar to ArcGIS ASCII grids, or other similar grid files, but in this case defines full 3D volume. Data is cell, i.e. each block is classified according to the stratigraphy. A table of colormaps identifying the classifications is created in xml colormap file. The data format comes from Geostatistics software package. In ParaView’s Plugin Manager, this plugin is mentioned as VoxelReader.
   - *License*: Owned and developed by Visual Technology Services Ltd.

3. **BandedContourFilter** – filter creates filled contours (banded cells) from the input poly-data,
   - *Description*: Filter plug-in creates filled contours poly-data from the input poly-data. Filled contours are bands of cells that all have the same cell scalar value and can therefore be colored alike. Used method is also referred to as a filled contour generation.
   - *License*: Owned and developed by Visual Technology Services Ltd.

4. **COMSOLReader** – reads mesh linear and quadratic elements from COMSOL files,
   - *Description*: Reader plug-in reads COMSOL mesh (.txt) and parameters (.txt) defined on some cloud of points (if available) files, calculates parameters values on the mesh elements using K-d tree data structure and outputs the resulting multi-block data. The currently supported elements are linear and quadratic.
   - *License*: Owned and developed by Visual Technology Services Ltd.

5. **ESRIFileReader** – reads .asc ASCII grid files with normal ArcGIS header,
   - *Description*: Reader plug-in reads ASCII grids (.asc) files with normal ArcGIS header. Cells must be square and NULL data must be marked with special value. Read data are outputted in uniform grid (vtkImageData) data structure.
   - *License*: Owned and developed by Visual Technology Services Ltd.
6. **Foam2ImageFilter** – filter creates an image with elevations attribute from the input multi-block data,

- **Description**: Filter plug-in creates an image data object with an elevation attribute array from the input multi-block surface. Filter works by performing the following 4 rules:
  - Pass through all points in output vtkImageData object and set their Elevation value to NaNValue value (previously set by user),
  - Pass through all the input triangles and for every triangle updates all points from the output vtkImageData object which are inside of the triangle's projection on the output vtkImageData object.
  - Under the point's update is performed the updating of per-point attribute array Elevation in which is kept the Z component of triangle's point with X, Y coordinates same as the point’s,
  - In case the point is initialized (the value is different from NaNValue), it is updated only if new Z coordinate is higher than previously stored. By using this technique, we force keeping the topmost part of the surface.

- **License**: Owned and developed by Visual Technology Services Ltd.

7. **GeoReferencedImage** – reads GeoTiff images supplied with world file,

- **Description**: Reader plug-in reads geo-referenced images (tif, tiff, jpg, jpeg, png) with a world file with the geospatial positioning data. Note, this plug-in uses the libtiff external library.

- **License**: Owned and developed by Visual Technology Services Ltd. The plug-in is derived from original open-source version from ParaViewGeo at http://paraviewgeo.objectivity.ca.

8. **GocadReader** – reads various Gocad file types tsurf, sgrid, voxet, etc,

- **Description**: Reader plug-in reads various Gocad files (.gp, .ts, .pl, .vs, .vts, .vo), tsurf, sgrid, voxet, etc. It doesn’t rely on the 3rd party library for the file reading, it fully implements reading of the supported Gocad files. Output object's type depends on the input types.

- **License**: Owned and developed by Visual Technology Services Ltd. The plug-in is derived from the original open-source version from ParaViewGeo at http://paraviewgeo.objectivity.ca.

9. **OBJFileExporter** – exports current 3D scene to Wavefront OBJ file format,

- **Description**: Exporter plug-in exports the current 3D scene to Wavefront OBJ file (.obj). It also exports the material file, allowing texture references to be added. Output of this exporter is suitable for the import into 3D visual simulation systems, or animation systems. This plug-in is also used from PDF3DParaViewplug-in when 3D scene is
exported to OBJ file(s). In this case, animation scene can be exported as a set of OBJ files or as OBJ + PC2 files.

- **License:** Owned and developed by Visual Technology Services Ltd. Derived from similar original open source VTK code.

10. **PLYFileExporter** – exports current 3D scene to Polygon File Format,

- **Description:** Exporter plug-in exports the current 3D scene to PLY file format (.ply). User can choose between binary and ASCII formats of output files. In case of animated scenes, every frame is exported to separate file with simple alteration in file-names by inserting frame-time, etc.

- **License:** Owned and developed by Visual Technology Services Ltd.

11. **Serafinplug-ins** – reads .slf, .res, .sel files, from fluid-dynamics simulation system Telemac,

- **Description:** Reader plug-in reads .srf, .slf, .res, .sel files, supporting the fluid-dynamics simulation system Telemac ([http://docs.opentelemac.org](http://docs.opentelemac.org)). The format can be 2D single triangle mesh layer or 3D prism volume file.

- **License:** The plug-in is originally from EDF, repackaged by HR Wallingford and modified by Visual Technology Services,

12. **ShapeFileReader** – reads ESRI ArcGIS shape file format (.shp), containing vector features and attributes,

- **Description:** Reader plug-in reads ESRI ArcGIS Shape files containing vector features and attributes. The current version loads both shapes and attributes.

- **License:** Owned and developed by Visual Technology Services Ltd. The plug-in is derived from the original open-source version from ParaViewGeo at [http://paraviewgeo.objectivity.ca](http://paraviewgeo.objectivity.ca).

13. **ZMAPPFileReader** – reads ZMap / ZCor 2D uniform grid data, defined by ASCII data similar to CSV file.

- **Description:** Reader plug-in which reads ZMap / ZCor 2D uniform grid data files (.dat, .grid, .zmap), defined by ASCII data similar to CSV file. The header contains some dimensional and geospatial data similar to ArcGIS ASCII grids. Data is 2D node array, a scalar normal depth, elevation, or some attribute.

- **License:** Owned and developed by Visual Technology Services Ltd.

14. **VisItBridge** – allows reading a large number of file types supported by VisIt.

- **Description:** Special readers subsystem allows using VisIt's IO components and exploring re-usable capabilities of VisIt and its underlying pipeline library avt. VisIt IO components are called the database plug-ins (they are not real ParaView's plug-ins, just a sub-system). Through the mechanism of the database plug-ins, it allows end-users to read a large number of file types. More information about the supported file-types can be found at [http://www.paraview.org/Wiki/VisIt_Database_Bridge](http://www.paraview.org/Wiki/VisIt_Database_Bridge).
• **License**: Owned and developed by Kitware Inc.
6 Working with Animations
ParaViewPlus® can create and read animated scenes, as well as exporting to various 3D PDF files formats.

1. Example of exporting animation with ParaViewPlus package

ParaViewPlus package is capable of exporting ParaView scenes into following file formats PDF, U3D (both pure and RHC), PRC (both pure and HCT), OBJ and PLY (both ascii and binary). For exporting animated scenes, it exports all frames into one PDF, U3D and PRC files. In case of exporting animated scenes to OBJ files, it is possible to export each frame to separate OBJ file, but also to one OBJ file and one PC2 (point cache) file which stores geometry changes (prerequisite is topology is constant throughout animation). Here are a few prerequisites before trying to export ParaView scene to one of supported formats:

- In case you want to export ParaView Scene to PDF, U3D or PRC file, the plug-in named PDF3DParaViewPlugin must be available and loaded into ParaView application.
- In case you want to export ParaView Scene to OBJ (+PC2) file, the plug-in named OBJFileExporter must be available and loaded into ParaView application beside the PDF3DParaViewPlugin.
- In case you want to export ParaView Scene to PLY file, the plug-in named PLYFileExporter must be available and loaded into ParaView application beside the PDF3DParaViewPlugin.
- You can manage loaded plug-ins using Plugin Manager window which is opened using Tools-Manage Plugins.

Turning ON animation export and setting up other animation exporting parameters can be set up on Animation tab in PDF3D Settings dialog, which is started using File-PDF3D Settings, shown in Figure 6.1.

![Figure 6.1: PDF3D Settings Dialog with Animation Parameters](image)

To enable exporting animation, Enable Animation Capture has to be turned ON. In this case, all frames with index Start Frame + i * Increment before End Frame will be exported.
• In case of exporting to OBJ, Export Point Cache (OBJ+PC2) turns on exporting animation scene to one OBJ file and one PC2 (point cache) file (contains series of geometry incremental changes, considering topology remained unchanged). In other case, frames will be exported to NumberOfFramesToBeExported files with file-names containing frame time and Time Unit Label.

• In case of exporting to PLY, Export Binary PLY (on Exporting tab) controls if created PLY files will be in Binary or ASCII file format. In this case, animation scene will be exported to NumberOfFramesToBeExported * NumberOfActorsPerFrame files with file-names containing frame time, Time Unit Label and actor name (i.e. Part number) in case there are multiple actors in the frame.

After animation exporting parameters are set up, 3D PDF file is exported by pushing PDF3D toolbar icon, selecting output file-name and output file type and pushing Save:

2. Achieving better compression performances

A common problem encountered when exporting animated scenes is resultant 3D PDF which are too large to be useful. This occurs because of two reasons:

• All animations (both camera and sequence animations) are exported as sequence (i.e. for every frame, entire mesh is exported separately)
• ParaView scenes may potentially be very large

Here are a few suggestions on how to get smaller 3D PDF files and better performances:

• Use simplification parameters on Conversion tab. Using this option, you can set output triangles / lines / points count targets in output 3D PDF file. Also, there are numerous other options for finer setting of simplification process. These parameters are applied per one-frame mesh, not globally,

• In case user has large number of streamlines, adjust number of seeding points for streamlines and stream tracer parameters like length, density of incremental advection points, etc. Exporting streamlines as tubes will provide a better quality of tessellation but carries additional computational overhead with more geometry than streamlines as line segments. PRC-HCT encoding is efficient in encoding triangles and can provide much smaller files streamlines are exported as tubes – trial and error is necessary

3. Advice for navigation between frames in 3D PDF files

After the animated 3D PDF file is created, users can navigate through different frames using animation widgets which are mostly exported under 3D scene. Using widgets users can start and stop animations, go to next frame, switch between different animation modes. Users may also enter frame number.
This software comprises unpublished confidential information including intellectual property of Visual Technology Services Ltd. and may not be used, copied or made available to anyone, except in accordance with the license under which it is furnished.