

VisIVOServer 2.0

User Guide

Introduction

Authors: Becciani U., Caniglia G., Costa A., Gheller C., Krokos M., Massimino P.

VisIVO Server is a suite of software tools for creating customized views of 3D renderings from astrophysical data tables. These tools are founded on the **VisIVO Desktop** functionality (visivo.oact.inaf.it) and support the most popular Linux based platforms (e.g. www.ubuntu.com). Their defining characteristic is that no fixed limits are prescribed regarding the dimensionality of data tables input for processing, thus supporting very large scale datasets.

VisIVO Server websites are currently hosted by the University of Portsmouth, UK (visivo.port.ac.uk), the INAF Astrophysical Observatory of Catania, Italy (visivo.oact.inaf.it) and in the near future by CINECA, Italy (visivo.cineca.it). These web sites offer data management functionality for registered users; datasets can be uploaded for temporary storage and processing for a period of up to two months. The sites can also be utilized through anonymous access in which case datasets can be uploaded and stored for a maximum of four days; to maximize available resources a limited dimensionality is only supported.

Assuming that datasets are uploaded, users are typically presented with tree-like structures (for easy data navigation) containing pointers to **files**, **tables**, **volumes** as well as **visuals**.

Files point to single, or possibly several (for distributed datasets), astrophysical data tables;

Tables are highly-efficient internal VisIVO Server data representations; they are typically produced from importing datasets uploaded by users using VisIVO Importer (see below);

Volumes are internal VisIVO Server data representations; they are produced either from direct importing of user datasets or by performing operations on already existing tables;

Visuals are collections of highly-customized, user-produced views of 3D renderings of volumes.

VisIVO Server consists of three core components: **VisIVO Importer**, **VisIVO Filter** and **VisIVO Viewer** respectively. Their functionality and usage is described in the following sections.

To create customized views of 3D renderings from astrophysical data tables, a two-stage process is employed. First, VisIVO Importer is utilized to convert user datasets into **VisIVO Binary Tables** (VBTs). Then, VisIVO Viewer is invoked to display customized views of 3D

renderings. As an example, consider displaying views from only three columns of an astrophysical data table supplied in ascii form, say col_1, col_2 and col_3, by using the commands

```
VisIVOImporter --fformat ascii UserDataSet.txt
```

```
VisIVOViewer -x col_1 -y col_2 -z col_3 --scale --glyphs pixel VBT.bin
```

VisIVOServer is distributed with GPL V.2 License for NON COMMERCIAL use. VisIVOServer is hosted by sourceforge <https://sourceforge.net/projects/visivoserver/> and its source code is downloadable via svn:

```
svn co https://visivoserver.svn.sourceforge.net/svnroot/visivoserver/branches/1.2 visivoserver
```

Disclaimer: user data integrity is never warranted.

VisIVO BINARY TABLE

A VisIVO Binary Table (VBT) is a highly-efficient data representation used by VisIVO Server internally. A VBT is realized through a header file (extension **.bin.head**) containing all necessary metadata, and a raw data file (extension **.bin**) storing actual data values. For example, the header may contain information regarding the overall number of fields and number of points for each field (for point datasets) or the number of cells and relevant mesh sizes (for volume datasets).. The raw data file is typically a sequence of values, e.g. all X followed by all Y values.

Header

The header file contains the following fields:

```
float | double
n1
n2 [ GeoX GeoY GeoZ DX DY DZ ]
little | big
X
Y
Z
Vx
Vy
Vz
```

- **float | double** is the data type of the storage variables used;
- **n1** denotes the number of columns (fields) in the VBT;
- **n2** denotes the number of rows in the VBT;

- **GeoX GeoY GeoZ DX DY DZ** are employed only if the VBT represents volumetric datasets. In that case GeoX, GeoY and GeoZ represent the mesh geometry, while DX, DY and DZ represent the x, y and z size of volumetric cells.
- **little | big** denotes the endianness employed in the VBT. After this field there exist $n1$ rows that indicate the VBT columns as positions (X, Y, Z) and velocities (Vx, Vy, Vz).

Raw

The binary file is simply a sequence of $n1*n2$ values. In the example shown in section 2.1 all X values, then all Y values and so on.

Note:

- **$n1$** represents the number of columns (fields) in the VBT (e.g. 6);
- **$n2$** represents the number of elements of each field in the VBT (e.g. 262144);
- **GeoX GeoY GeoZ** represent the number of the volumetric cells in each dimension of the mesh size (used only for Volumes) (e.g. 64 64 64);
- **DX DY DZ** represent the size of each cell (used only for Volumes) (e.g. 1.0 1.0 1.0)

VisIVO Server 1.2

VisIVO Utilities

VisIVO Utilities is a tool that creates intermediate data that will be used by the other components of VisIVO Server. These data could consist of a sequence of values or files extracted from VisIVO Binary Tables.

GENERAL SYNTAX

VisIVOUtils --help
produces a general help

VisIVOUtils --op utilitycode --help
produces the help of a specific utility

VisIVOUtils --op utilitycode <options>
runs the utility

The following utilities are available

Create path

append or creates an ascii file containing three values for each row with azimuth elevation and zoom

Orthogonal Slices

append or creates an ascii file containing the slice position in a volume: used in VisIVOViewer --cycle option for *slice* visualization.

Generic Slices

append or creates an ascii file containing a plane point and the normal point for plane: used in VisIVOViewer with --cycle option for slice visualization.

VisIVO Utilities

Create path

This utility append or creates an ascii files containing 4,7 or 8 values for each row for the camera position. The file can be used by VisIVOViewer --cycle to produce a sequence of png images to be mounted in a movie.

The camera position, focal point and roll, when not specified, will contain NULL that will allow VisIVO Viewer to maintain the last used setting.

Usage:

VisIVOUtils --op createpath --type value [--azimuth from to] [--elevation from to] [--zoom from to] [--zoomend [stepframe]] [--campos from to] [--camfp from to] [--camroll from to] [--framesec value] [--length value] [--out filename] [--help]

Example:

VisIVOUtils --op createpath --type 0 --azimuth 0.0 60. --elevation 0.0 10.0 --zoom 1.0 1.5 --zoomend --length 20 --out my_cycle.par

The utility produce 10 values (default value) for each second. The file my_cycle.par will contain: azimuth, elevation, zooming and roll:

```
0.0  0.0  1.0 NULL
.....
60.0 10.0  1.0 NULL
60.0 10.0  1.2 NULL
60.0 10.0  1.4 NULL
60.0 10.0  1.6 NULL
```

VisIVOUtils --op createpath --type 1 --azimuth 0.0 60. --elevation 0.0 10.0 --camfp 35 35 35 20 20 --zoom 1.0 1.5 --zoomend --camroll 0 0 --length 20 --out my_cycle.par

The utility produce 10 values (default value) for each second. The file my_cycle.par will contain: azimuth, elevation, zooming, camera focal point and roll:

```
0.0  0.0  1.0 35 35 35 0
.....
60.0 10.0  1.0 20 20 20 0
60.0 10.0  1.2 20 20 20 0
60.0 10.0  1.4 20 20 20 0
60.0 10.0  1.6 20 20 20 0
```

VisIVOUtils --op createpath --type 2 --campos 35 35 200 35 35 70 --camfp 35 35 35 20 20 20 --length 20 --out my_cycle.par

The utility produce 10 values (default value) for each second. The file my_cycle.par will contain: zoom, camera position, camera focal point and roll:

```
1.0 35 35 200 35 35 35 NULL
.....
1.0 35 35 70 20 20 20 NULL
```

VisIVOUtils --op createpath --type 2 --campos 35 35 200 35 35 70 --camroll 0 60 --length 20 --out my_cycle.par

```
1.0 35 35 200 NULL NULL NULL 0
.....
1.0 35 35 70 NULL NULL NULL 60
```

VisIVOUtils --op createpath --type 3 --azimuth 0.0 60. --elevation 0.0 10.0 --campos 35 35 200 35 35 200 --camfp 35 35 35 35 35 35 --zoom 1.0 1.5 --length 20 --out my_cycle.par

The utility produce 10 values (default value) for each second. The file my_cycle.par will contain: zoom, camera position, camera focal point and roll:

```
0 0 1.0 35 35 200 35 35 35 NULL
.....
60 10 1.5 35 35 200 35 35 35 NULL
```

It is suggested to use this type to move the camera on azimuth, elevation and zoom only, putting fix the other values

Note:

--type . 0 Create path for azimuth, elevation, zoom and roll. Default value.
 1 Create path for azimuth, elevation, zoom, focal point and roll
 2 Create path for zoom, camera position, focal point and roll
 3 Create path for azimuth, elevation, zoom, camera position, focal point and roll

--azimuth Movement *from to*. Default values 0.0 and 0.0.

--elevation Movement *from to*. Default values 0.0 and 0.0. Valid range [-90,90]. Values outside this interval are automatically set to the near extreme: Ex.: *--elevation -85 100* will be modified with *--elevation -85 90*.

--zoom Zoom *from to*. Default values 1.0 and 1.0. A zooming factor <1 represents a zoom in a zooming factor >1 represent a zoom out. Negative value are ignored

--zoomend The zoom is given at the end. The value step-frame represent the step for zooming . Default step-frame is 0.2 If this option is given priority with zoom will be ignored. The final zooming is added to global the length.

--campos Movement *from to*. Three vale for starting point and three value for ending point are expected.

--camfp Movement from to. Three value for starting point and three value for ending point are expected.
--camroll movement from to. To values starting and ending degrees are expected
--framesec Number of frame values for each second. Default value is 10 .
--length Value in seconds. Default value is 10 sec.
--out Output filename. Default filename cycle.par. The file is opened in append mode.

VisIVO Utilities

Orthogonal Slices

This utility append or creates an ascii file containing the slice position in the volume table. The file can be used by VisIVOViewer --cycle to visualize slices and to produce a sequence of png images to be mounted in a movie.

Usage:

VisIVOUtils --op orthoslices --pos from to [--xplane] [--yplane] [--zplane] [--step stepvalue] [--out filename] [--help] [--file inputFile.bin]

Example:

VisIVOUtils --op orthoslices --pos 0 64 --step 1 --out my_cycle.par --file inputFile.bin

The utility produces 64 values as follows:

0
1
2
...
64

Note:

--pos Sets the slice position from-to in the volume. Values outside the volume size are ignored.
--xplane Sets the direction x to be considered. Default is x
--yplane Sets the direction y to be considered. It is ignored if --xplane is given
--zplane Sets the direction z to be considered. It is ignored if --xplane or --yplane is given
--step Step increment for slice position (integer). Default value 1
--out output filename. Default filename cycle.par. The file is opened in append mode
--file (optional) Input Volume table

VisIVO Utilities

Generic Slices

This utility append or creates a file with six columns. The point position (plane point) has increased (decreased) of `step_size` for `n` steps. The plane point is moved along the normal axis. The product ***step*size*** determines the movement of the plain point.

If *step*size* is equal to 1, at the end the plane point will be at the same point of the normal point.

The file can be used by VisIVOViewer `--cycle` to visualize generic slices and to produce a sequence of png images to be mounted in a movie.

Usage:

VisIVOUtils `--op genericslices --point x y z --normal x y z --step n [--size step_size] [--movedown] [--out filename] [--help]`

Example:

VisIVOUtils `--op genericslices --point 1 1 1 --normal 10 10 10 --step 20 --size 0.05 --out cyclefile`

The utility produces 21 values (including the starting point) as follows:

```
1 1 1 10 10 10
1.45 1.45 1.45 10 10 10
1.9 1.9 1.9 10 10 10
2.35 2.35 2.35 10 10 10
2.8 2.8 2.8 10 10 10
3.25 3.25 3.25 10 10 10
3.7 3.7 3.7 10 10 10
4.15 4.15 4.15 10 10 10
4.6 4.6 4.6 10 10 10
5.05 5.05 5.05 10 10 10
5.5 5.5 5.5 10 10 10
5.95 5.95 5.95 10 10 10
6.4 6.4 6.4 10 10 10
6.85 6.85 6.85 10 10 10
7.3 7.3 7.3 10 10 10
7.75 7.75 7.75 10 10 10
```



```
8.2 8.2 8.2 10 10 10
8.65 8.65 8.65 10 10 10
9.1 9.1 9.1 10 10 10
9.55 9.55 9.55 10 10 10
10 10 10 10 10 10
```

Note:

- point** The three coordinates of a point in the plane.
- normal** The three coordinates fixing the normal axis to the plane
- step** Number (int) of generated new point positions along the normal axis
- size** Value of increased (decreased) point coordinates. Default value 1
- movedown** The plane point is moving to the opposite side of the normal point
- out** output filename. Default filename cycle.par. The file is opened in append mode