VisIVOServer 2.1

User Guide

Introduction

Authors: Becciani U., Bandieramonte M., Costa A., 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;

- ***Geo*X 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 endianism 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)

VisIVOServer

VisIVO Filters

VisIVO Filters are programs that convert data from an original data table (internal data format table) into a new one or can create a Volume from a table. Filters are directly applied by the user using VisIVO Server Website: the web procedure directly calls the filters when applicable.

GENERAL SYNTAX:

VisIVOFilters --help
produces a general help

VisIVOFilters --op operation --help
produces the help of a specific operation

VisIVOFilters --op operation <options> [--file] InputFile
runs the operation

VisIVOFilters parameterFile
runs the operation with options specified in the parameterFile

Where requested the InputFile is a valid table (Binary Internal data format). In this case the *InputFile.bin* and *InputFile.bin.head* must exist.

GLOBAL OPTIONS:

The following option can be given on any filter operation.

--memsizelimit percentage_value

This option reduces the memory request of the percentage value given with this option. Depending on the specific filter request and on the system where VisIVOFilters runs, the allocated memory could exceed the available size and the application could be aborted or can use a significant portion of the system swap area, with a dramatic lost of performance. This parameter can be given to reduce the allocated space avoiding this effect.
A Warning message will be given when this option is used. The allowed value is a float greater than 0. and lower than 95.0.

NOTE: The filters AHFstep, AHF Galaxy Extraction, AHF Halos List and VBT2AHF are available only if the AHF option is used when compiling VisIVO

The following filter operations are available

Add Identifier
adds a column tag to a table

AHFstep
runs the serial version of Amiga Halofinder (v0.0/271)

AHF Galaxy Extraction
extracts up to three multi-lists from a multi-list. Designed for "particles" files of Amiga

AHF Halos List
extracts a multi-list from a multi-list. Designed for "particles" files of Amiga

Append Tables
creates e new table appending data from a list of existing tables

Cartesian2polar
creates three new fields as the result of the polar transformation.

Change Column Name
changes the column names in a table.

Coarse Volume
produces a coarse sub-volume with plane extraction from the original volume

Cut
sets to a given threshold, column values included in a given interval.

Decimator
creates a sub-table as a regular subsample from the input table.

Extract List
creates a new table with the element of input table listed in a separate file.

Extract Subregion
creates a new table from an input table of a sub-box or of a sphere.

Extract Subvolume
produces a table which represents a sub-volume from the original volume.

Grid2Point
distributes the values of an input grid to a point distribution in the same domain using the NGP/CIC/TSC algorithm

Interpolate
creates new tables from two existing data tables with a linear interpolation.

Include
assigns a different offset value to points inside and/or outside a sphere in the domain.

Math. Operations
creates a new field in a data table as the result of a mathematical operation between the existing fields.

Merge Tables

creates a new table from two or more existing data tables.

Module

`creates a new table field` computing the module of three fields of the input table.

Multi-resolution

creates a new table using different levels of randomization inside a dataset.

Poca

creates a set of pointsand a volume starting from data plane. Support to the Muon portal Project.

Point Distribute

creates a volume using a field distribution (CIC/NGP/TSC algorithm) on a regular mesh.

Point Property

assigns a property to each data point on the table

Randomizer

creates a random subset from the original data table.

Select Columns

creates a new table using one or more columns of a data table.

Select Rows

creates a new table using limits on one or more fields of a data table.

Show Table

produces an ASCII table of a selected field of the first number of rows (or of all rows).

Show Volume

writes, on an output ascii file, the volume cells and their values that satisfy given limits

Sigma Contours

creates a new table where one or more fields of a data table have values within N sigma contours

Statistic

creates and plots an histogram of a scalar field of a data table.

Split Table

splits an existing table into two or more tables

Swap

swaps a table from big endian to little endian and viceversa.

VBT2AHF

converts a table into the Amiga Halo Finder data file format.

Visual

creates a randomized new table from an input table listing tables and columns with equal data size, to be used with VisIVOViewer.

Write VOTable

creates a VOTable using schema 1.2 from an input VBT

NOTE: from the subversion 1.1 VisIVOFilters incapsulate the Amiga Halo Finder code v0.0/271 (serial version only). It also includes the vbt2ahf filter that converts a VBT into the

amiga data file format. This data format can be used to execute the filter AHFstep or externally to execute the Amiga code

VisIVO Filter on gLite systems

VisIVO may be compiled on a gLite system. In this case the option *GLITE* must be given to compile the tool. The main purpose is to read a VBT from catalogue and/or create a modified VBT in the catalogue.
When running on gLite system:

the option –-VO is to specify the virtual organization (VO) name.

2) the input VBT file, to be imported, can be a logical filename (lfn) and must start with *lfn:// .*

3) the output VBT, can be local (-- out option) or the output lfn can be given. But in any case the –-out is the local filename where a temporary VBT will be created

4) - -lfnout: is the output logical filename. In this case a VBT will be saved in the grid catalogue and deleted from local filesystem.

5) the option –-se is to specify the Storage Element (SE)

The following options can be applied to all filters:

--VO [value] (optional) is used to set the virtual organization (VO) when running on gLite grid. It is mandatory when running on gLite using catalogue.
--se [value] (optional) is used to set the Storage Element (SE) when running on gLite grid. Default value is DPM_HOST (site default storage element).
--lfnout [value] (optional) is used to set the logical filename (lfn) when running on gLite grid.

VisIVO Filters

ParameterFile

All the following listed filters contain the –-op code and options. The code and all the options can be given in a parameterFile

Lines starting with # are comments.

An example of this file is the following (for the mathop filter):

```
op=mathop                    ← a valid filter code
#memsizelimit=30             ← this is a commented line
#expression=myexp.txt        ← this is a commented line
compute=(A*B)/C              ← No  escape and special "<<,>>" characters mut be given
append=true
outcol=NewColName
#out=outFilename             ← this is a commented line
#help=true                   ← this is a commented line
file=inputFile.bin
```

An example of this file is the following (for the pointproperty filter):

```
op=pointproperty             ← a valid filter code
#memsizelimit=30             ← this is a commented line
resolution= 16 16 16
points= X Y Z
field=mass
append=true
periodic=true
outcol=NewColName
#out=outFilename             ← this is a commented line
#help=true                   ← this is a commented line
file=inputFile.bin
```

NOTE:

1) The parameterFile must contain the options with the same names reported in the following as "--" options.
2) Options with one or more parameters must be all specified after the "=" sign in the same line. (ex. `field=X Y Z`).
3) Options that do not require parameters must be given with "true" keyword (ex. `append=true`).

VisIVO Filters

Add Identifier

This filter adds a new column with a sequence of Ids in the input data table.
Usage:

VisIVOFilters --op addId [--outcol col_name] [--start start_number] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op addId --outcol PtId --file inputFile.bin

The command produces a new column named PtId: a sequence of tags starting from 0.

Note:
--start Starting Id. Default value is 0. Only an int value can be given.
--outcol. Column name of the new id column. Default name is Id.
--file Input table filename.

VisIVO Filters

AHFstep

This filter executes the serial version of Amiga Halofinder (v0.0/271). See the Amiga HF documentation for more details.

NOTE: The filters is available only if the AHF option is used when compiling VisIVO.

Usage:

VisIVOFilters --op AHFstep [--help] --input AmigaParameterFile

Example:

VisIVOFilters --op AHFstep --input myparFile.par

The command produces the following file

prefix.halos, *prefix*.mass, *prefix*.particles, *prefix*.profiles, *prefix*.substructure

containing the halos properties computed by the inputfile

Note:
--input. A parameter file that has the following fields.

file with initial conditions (filename in the Amiga halo finder data format. See vbt2ahf filter)
prefix for output file names (filename prefix)
number of domain grid cells (1D)
Nth for domain grid refinement criterion
(= number of particles per cell)
on the domain grid
(0-6 suggested values)
Nth for refinements refinement criterion
(= number of particles per cell)
on all refinement grids
(0-6 suggested values)
final redshift (unimportant value for AHFStep)
dump step (unimportant value for AHFStep)
total number of output files (unimportant value for AHFStep)

Example:

out_8Ml_0.3000_amg
AMG8ML_out_z0
128
4
4
0
0
0
0

VisIVO Filters

AHF Galaxy Extraction

This filter is designed for Amiga *particles* file. This file is a multilist of particles in halos. The filter extracts particles and creates up to three multi-lists.

It extracts up to three ascii multi-lists from a general input ascii multilist, following the Ids in a given input table, and using the threshold value given in the select value.

NOTE: The filters is available only if the AHF option is used when compiling VisIVO.

Usage:

VisIVOFilters --op ahfhalogalaxyext --field col1 col2 col3 --multilist in_mutlist_File [--threshold value] [--out out_multilist_file] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op ahfhalogalaxyext --field Id1 Id2 Id3 --multilist input.AHF_particles --out my_multilist.lst --file sub_AHF_halos.bin

The lists are extracted: my_multilist.lst_Id1_particles where Id1 >= 1.0, my_multilist.lst_Id2_particles where Id2 >= 1.0 and Id1 <1.0, my_multilist.lst_Id3_particles where Id3 >= 1.0 and Id1 <1.0, Id2<1.0, from input.AHF_particles. The threshold value 1.0 (default) can be changed with the specific option. Id1,Id2 and Id3 must be valid column name.

Note:
--field. Up to three column names. Default name is Id.
--multilist. Ascii filename with a multilist.
--threshold. Selector value among the three columns. Default Value 1.0
--out. Output ascii root-filename with a multilist.
--file Input table filename.

VisIVO Filters

AHF Halos List

This filter is designed for Amiga *particles* file. This file is a multi-list of particles in halos. The filter extract some lists and creates a new multi-list. It reads a column from an input table with Ids (integer positive values) that represents the list to be extracted from the input multi-list file.

NOTE: The filters is available only if the AHF option is used when compiling VisIVO.

Usage:

VisIVOFilters --op ahfhalolist [--field col_name] --multilist in_mutlist_File [--out out_multilist_file] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op ahfhalolist --field Id --multilist input.AHF_particles --out my_multilist.lst --file sub_AHF_halos.bin

The command produces the *my_multilist.lst* file that has similar structure to the AHF_particles file produced by Amiga

Note:
--field. Column name. Default name is Id.
--multilist. Ascii filename with a multilist.

--out. Output ascii filename with the extracted multilist.
--file Input table filename.

VisIVO Filters

Append Tables

This operation creates e new table appending data from a list of existing tables. Append Filter can append up to 100 tables with the same number of Columns

Usage:

VisIVOFilters --op append [--out filename_out.bin] [--help] [--filelist] table_list.txt

Example:

VisIVOFilters --op append --out out_table.bin --filelist tab_list.txt

tab_list.txt is a file that contains a list of valid table names. The ".bin" extension is automatically added if the listed filename does not contain it.

/home/user/tab1.bin
/home/user/tab2.bin
...
/home/user/otherDirectory/tabN.bin

The filter produces a new table (*out_table.bin* and *out_table.bin.head*). The column names are copied from the first table. An error is given if tables contain different numbers of columns.

Note:
--out Name of the new table. Default name is given.
--filelist (obslote --file) An ascii file with a valid list of tables

VisIVO Filters

Cartesian2polar

This operation creates three new fields in a data table as the result of the spherical polar transformation of three existing fields.

Usage:

VisIVOFilters --op cartesian2polar --field X Y Z [--append] [--outcol rho theta phi] [--out filename_out.bin] [--file] inputFile.bin

Example:

VisIVOFilters --op cartesian2polar --field X Y Z --append --file inputFile.bin

This command appends three new fields to the inputFile.bin file that represent s the spherical polar transformation.

Note:
--field Three valid columns name used as cartesian coordinates
--append No new table will be created. The original table will have new fields. Default options: a new table with only the new field is produced.
--outcol. Column name of the new fields. Default names are: rho, theta and phi.
--out Name of the new table. Default name is given. Ignored if *--append* is specified.
--file Input table filename

VisIVO Filters

Change column names

This operation changes the column names in an existing table.

Usage:

VisIVOFilters --op changecolname --field column_names --newnames new_names [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op changecolname--field X_r Y_r --newnames X_n Y_n --file inputFile.bin

This command changes the column names X_r into Xn and Y_r into Y_n in the inputFile table

Note:
--field Valid columns names
--newnames Valid new columns names.
--file Input table filename

VisIVO Filters

Coarse Volume

This operation produces a coarse sub-volume with plane extraction from the original volume.

Usage:

VisIVOFilters --op coarsevolume [--perc percentage] [--newres x_res y_res z_res] [--field column_names] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op coarsevolume --perc 10.0 --field Mass Temperature --out subvolume.bin --file inputFile.bin

produces a new table (*subvolume.bin* and *subvolume.bin.head*). It is a volume of 10 percent of the original volume. If the original volume is a mesh of 320x320x640 the output volume will be 32x32x64 and only the listed fields Mass and Temperature will be considered. Planes are extracted from the original volume as a regular distribution from the original mesh.

Note:
--perc A percentage (from 0.0 to 100.0) sub-volume will be produced. Default value produces a sub volume that could be directly uploaded and visualized with VisIVOViewer and VisIVODesktop applications.
--field List of columns contained in the original file. Default: all columns will be extracted.
--newres A sub-volume with new resolution will be produced. No default is given. This parameter is ignored if *--perc* option is given
--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Cut

This operation fixes to a threshold, column values included in an interval.

Usage:

VisIVOFilters --op cut [--field columns_list] --limits filename_limits [--threshold value] [--operator AND/OR] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op cut --field A B C --limits limitsfile.txt --operator AND --out filename_out.bin --threshold 1.0 --file inputFile.bin

The limitsfile.txt file must have the following structure. A valid column name and an interval indicating the extraction limits:

X 20.0 30.0
Y 10.0 20.0
Z 0.0 10.0

The command produces a new table (*filename_out.bin* and *filename_out.bin.head*) that contains all the data points and columns of the inputFile table. In any row where *X*=[20.0,30.0] AND *Y*=[10.0,20.0] AND *Z*=[0.0,10.0] the fields A B and C will be changed with the threshold value 1.0. Other fields will not be changed. The **unlimited** word can be used to indicate the infinite value.

Note:
--field it is a valid columns name list to be reported in the new table. Default: all columns will be reported.
--limits A file that has three columns: a valid column name and an interval indicating the extraction limits.
--threshold Value to be used to cut data. Default value is 0.0
--operator Limits on all fields listed in the *--limits* option file are combined by default with logic AND operator. If this option is given with the OR value the field limits are combined with logic OR operator
--out Output table filename. Default name is given.
--file Input table filename.

VisIVO Filters

Decimator

This operation creates a sub-table as a regular subsample from the input table.

Usage:

VisIVOFilters --op decimator --skip step [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op decimator --skip 9 --out inputFile_samp.bin --file inputFile.bin

produces a new table (*filename_out.bin* and *filename_out.bin.head*) having a 10 percent size of the original inputFile. Values are extracted in a regular sequence, skipping step element every time. The skip value is an integer number > 1 and represents the number of skipped values. In the above example only one element every 10 elements will be reported in the output file. The output table must fit the available RAM.

Note:
--skip Integer representing the skipped values

--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Extract List

This operation creates a new table from an input table with the elements (rows) listed in a given multi-list file.
The filter extracts rows from a data table. A multi-list is given in ascii or binary format (unsigned long long int).
The multi-list has the following structure:

Number NL of lists
NL sequences of
1. Number N0 of elements in the list
2. N0 element ids (numbers of rows)

Option can be given to provide the NL number. In this case the multi-list file must not contain this information.
Option can be given to provide the N0 number. In this case the multi-list file must not contain this informations but it is a multi-list, each list must contain N0 elements.
If the onelist option is given the multi-list file is only a sequence of rows to be extracted.

Multi-list ascii file example

```
10          <= Number of lists (must not be given if the numberlists option is specified)
50          <= Number of elements of the first list
                  (must not be given if the listelemnts option is specified)
1           <= Sequence of rows (50) forming the list
7
273
1005
23
.....

35    <= Number of elements of the second list
2     <= Sequence of rows (35) forming the list
1006
145
15
.......
etc.
```

Operation not allowed on volumes.

Usage:

VisIVOFilters --op extractlist --multilist filename_list [--binaryint] [--asciilist] [--numberlists nl] [--listelements n0] [--onelist] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op extractlist --multilist mylist.txt --asciilist --out list_extract.bin --file inputFile.bin

The multilist.txt file has the following structure:

Number NL of lists
NL sequences of
1. Number N0 of elements in the list
2. N0 element ids (numbers of rows)

Multi-list ascci file example

```
10          <= Number of lists (NL)
50          <= Number of elements (N0) of the first list
1           <= Sequence of rows (N0=50) forming the list
7
273
1005
23
.....

35    <= Number of elements (N0) of the second list
2     <= Sequence of rows (N0=35) forming the list
1006
145
15
.......
etc.
```

The command produces a new table (*list_extracted.bin* and *list_extracted.bin.head*) that contain all data rows of the inputFile table (all columns will be reported) that are included in the multi-list file

Note:

--multilist is the multi-list file name.
--binaryint The default multi-list file format is binary unsigned long long int. If this parameter is specified the file is binary int.
--asciilist The multi-list file format is binary. If this parameter is specified the file is an ascii text.
--numberlists The multi-list file format is just a sequence of nl lists specified in this option. Each list starts with the number of elements in the list

--listelements The multi-list file format is just a sequence of nl lists. Each list has the same number of n0 elements. This option requires that the *--numberlists* option is specified, otherwise it is ignored.
--onelist If this option is given, the multi-list file is considered as only one list. Each element is the ID of the particle to be extracted. The *--numberlists* and *--listelements* options will be ignored.
--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Extract Subregion

This operation creates a new table from a sub-box or a sphere. Operation not allowed on volumes.

Usage:

VisIVOFilters --op extraction --geometry geometry_file [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op extraction --geometry geometry.txt --out pos_extracted.bin --file inputFile.bin

The geometry.txt file must have four rows and two columns. The first three rows have a valid column name and a value that indicates the extraction coordinates:

X 20.0
Y 20.0
Z 20.0
CORNER 10.0

The command produces a new table (*pos_extracted.bin* and *pos_extracted.bin.head*) that contain all data points of the inputFile table (all columns will be reported) that are included in the box having the lower corner at X=20.0 Y=20.0 and Z=20.0 and size=10.0

Note:
--geometry is a file name that contains three valid column names and a value for each column. The fourth field means the extraction mode and the sub-volume size:

RADIUS: a sphere centered in the given values will be extracted
CORNER: a rectangular region having the lower corner at the given values will be extracted
BOX: a rectangular region centered in the given values will be extracted.

Examples:

X 25.0
Y 25.0
Z 25.0
RADIUS 5.0

or

X 25.0
Y 25.0
Z 25.0
BOX 5.0

or

X 0.0
Y 0.0
Z 0.0
CORNER 10.0

--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Extract Subvolume

This operation extract a table which is a sub-volume from the original volume

Usage:

VisIVOFilters --op extractsubvolume --startingcell X Y Z --resolution x_res y_res z_res [--field column_names] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op extractsubvolume --startingcell 8 8 8 --field Mass Temperature --resolution 16 16 16 --out mysubvolume.bin --file inputFile.bin

produces a new table volume (*mysubvolume.bin* and *mysubvolume.bin.head*) that is a sub-volume of resolution 16x16x16 from the original volume and starting from the cell (8,8,8) of the original mesh. Only Mass and Temperature fields will be reported in the new table.

Note:
--startingcell X Y Z number of the first cell to be extracted: 0 0 0 is the first cell of the original grid

--resolution Grid size (3D) of the new subgrid
--field Valid columns name list to be reported in the new table.
--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Grid2Point

This operation distributes a volume property to a point data set on the same computational domain using a field distribution (CIC/NGP/TSC algorithm) on a regular mesh. CIC is the default adopted algorithm. The Cell geometry is considered only to compute the cell volume value in this operation.
This filter produces a new table or adds a new field to the input table. The operation performs the following:

1) It loads a volume (input volume data table) and a table with a point distribution in the same volume

2) It computes, using a CIC or NGP or TSC algorithm, a value (assumed density) for each data point, considering the cells value where the point is spread. The grid points density values are multiplied for the cell volume and assigned to the point. If the density option is given the cell volume is assumed =1;

3) It saves the property in a new table or adds the field to the original input table.

Usage:

VisIVOFilters --op grid2point --points x_col y_col z_col [--field column_name] [--density] [--append] [--out filename_out.bin] [--outcol col_name] [--tsc] [--ngp] [--help] --volume inputVolmeData.bin [--gridOrigin xg0 xg1 xg2] [--gridSpacing sg0 sg1 sg2] [--box length] [--periodic] [--file] inputFile.bin

Example:

VisIVOFilters --op grid2point --points X Y Z --field Mass --append --volume myvolume.bin --outcol distribute --file inputFile.bin

It adds a new column *distribute* to the inputFile.bin table. The filter distributes the *Mass* field of myvolume.bin VBT to the points *X Y Z* using a CIC algorithm

Note:

--points Columns to be assumed for points coordinates.
--field Valid Volume Column Name. Default value is the first column name

--density. Cell volume is not considered (cell volume=1)
--append. No new table will be created. The original table will have the new field.
--out Name of the new table. Default name is given. Ignored if --append is specified.
--outcol. Column name of the new field
--tsc. The TSC algorithm is adopted.
--ngp. The NGP algorithm is adopted.
-**-volume**. Input data volume filename (a VisIVO Binary Table).
--gridOrigin. It specifies the coordinate of the lower left corner of the grid. Default values are assumed from the box of inputFile.bin
--gridSpacing. It specifies the length of each cell dimension in arbitrary unit. This parameter is ignored if the box option is given. Default values are assumed from the box of inputFile.bin
--box. It specifies the length of a box. Default value is assumed from the box of inputFile.bin if the gridSpacing option is not given
--periodic applies a periodical boundary condition
--file Input table filename with point distribution.

VisIVO Filters

Interpolate

This operation creates new tables from two existing data tables (mainly used to produce intermediate frames of a dynamical evolution).

Usage:

VisIVOFilters --op interpolate [--field columns_name] [--numbin numberbin] [--periodic] [--interval from to] [--out filename_out] [--help] --infiles file_start.bin file_end.bin

Example:

VisIVOFilters --op interpolate --field X Y Z --numbin 20 --periodic --out mysequence --infiles start.bin end.bin

This command produces new tables (mysequence_0.bin, mysequence_1.bin .... mysequence_19.bin and the header files) having only the three fields X, Y and Z. This Filter creates a file in the same directory of the -–*out* file VSInterpolatefilelist_time.txt that lists all names on created files, where time is the current time.

Limits:
The infiles tables must have the listed columns in the --field option in the same corresponding order. If Y is the second column of the start.bin file, the end.bin file must contain Y as the second column. The input tables must have the same number of rows and the interpolated elements are considered in the same order. No index is currently supported.

Note:
--field (obsolete --list) a valid list of columns names that must exist on both input tables. Default: all columns in infile files are considered.

--numbin is the number of bins between the file_start.bin and the file_end.bin input files or the interval given in the --interval option. The default value is 10. The number of created tables is equal to numberbin-1.
--periodic applies a periodical boundary condition
--interval. VisIVO assumes a distance of 1.0 between the two input frames: file_start.bin and file_end.bin. This option produces the intermediate frames (tables) in a subinterval between the two input frames. The value 0.5 is the medium point of the interval. If the from value is lower than 0.0 it is considered 0.0. If the to value is lower than 1.0 it is considered 1.0. If the from value is equal to to value the operation is not performed. Default value from=0.0 to =1.0
--out is the root name of the new tables. The default name is given. The new name is given by the filename_out#.bin where # is the number of created tables.
--infiles contains the names of the input tables of the interpolation process.

VisIVO Filters

Include

This operation produces a new table or adds a new field to the input table. Points inside the sphere (given with center and radius) will have the value *invalue*, otherwise *outvalue*..

Usage:

VisIVOFilters --op include --center x_coord y_coord z_coord --radius radius [--field x_col y_col z_col] [--append] [--out filename_out.bin] [--outcol col_name] [--outvalue outvalue] [--invalue invalue] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op iinclude --center 1.0 1.0 1.0 --radius 1.0 --field X Y Z --append --outcol INOUT [--help] --file inputFile.bin

This command creates a new column in the input table. Points inside the sphere (center and radius are given) will have the default *invalue* equal to 1.0. Points outside the sphere will have the default *outvalue* equal to 0.0.

Note:
--center Coordinates of the sphere center.
--radius Radius of the sphere.
--field Three valid columns names. Default values are the first three columns.
--append No new table will be created. The original table will have the new field.
--out Name of the new table. Default name is given. Ignored if --append is specified.
--outcol Column name of the new field
--outvalue Value given to points outside the sphere. Default value is 0.
--invalue Value given to points inside the sphere. Default value is 1.
--file Input table filename.

VisIVO Filters

Mathematical Operations

It is based on Function parser for C++ v2.83 by Warp (http://iki.fi/warp/FunctionParser/) with some minor modifications. The filter creates a new field in a data table as the result of a mathematical operation between the existing fields

Usage:

VisIVOFilters --op mathop [--expression math_expression.txt] [--compute <<expression>>] [--append] [--outcol col_name] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op mathop --expression my_expression.txt --outcol SqrMod --out SqrModTable.bin --file inputFile.bin

my_expression.txt is a file that contains only one row with a mathematical expression:

*sqrt(VelX*VelX+VelY*VelY+VelZ*VelZ)*

being *VelX VelY* and *VelZ* columns name listed in *inputFile.bin.head*

The command produces a new table (*SqrModTable.bin* and *SqrModTable.bin.head*) with only one column called SqrMod having the same length of the columns of the inputFile.bin table

Note:
A new field is produced.
--expression A file with only one row having any valid mathematical expression with Valid Column names. Ignored if compute option is given.
--compute A valid mathematical expression with Valid Column names. The expression must start with << and finish with >> characters. It has the priority on the expression option. The expression must contain the escape character control for the << and >> symbols and the parentheses: To evaluate (A/B) *C the correct syntax will b –-compute \<\<\(A/B\)*C\>\> . NOTE: the << , >> and escape characters MUST NOT BE GIVEN if the parameter file is used.
--append No new table will be created. The original table will have the new field. Default options: a new table with only the new field is produced.
--outcol. Column name of the new field
--out Name of the new table. Default name is given. Ignored if --append is specified.
--file Input table filename

As reported in the Function parser library, function string is very similar to the C-syntax. Arithmetic float expressions can be created from float literals, variables or functions using the following operators in this order of precedence:

()	expressions in parentheses first
A unit	a unit multiplier (if one has been added)
A^B	exponentiation (A raised to the power B)
-A	unary minus
!A	unary logical not (result is 1 if int(A) is 0, else 0)
A*B A/B A%B	multiplication, division and modulo
A+B A-B	addition and subtraction
A=B A!=B A<B A<=B A>B A>=B	comparison between A and B (result is either 0 or 1)
A&B	result is 1 if int(A) and int(B) differ from 0, else 0
A|B	result is 1 if int(A) or int(B) differ from 0, else 0

Since the unary minus has higher precedence than any other operator, for example the following expression is valid: x*-y

The comparison operators use an epsilon value, so expressions which may differ in very least-significant digits should work correctly. For example, "0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1 = 1" should always return 1, and the same comparison done with ">" or "<" should always return 0. Without epsilon this comparison probably returns the wrong value.

abs(A)	Absolute value of A. If A is negative, returns -A otherwise returns A.
acos(A)	Arc-cosine of A. Returns the angle, measured in radians, whose cosine is A.
acosh(A)	Same as acos() but for hyperbolic cosine.
asin(A)	Arc-sine of A. Returns the angle, measured in radians, whose sine is A.
asinh(A)	Same as asin() but for hyperbolic sine.
atan(A)	Arc-tangent of (A). Returns the angle, measured in radians, whose tangent is (A).
atan2(A,B)	Arc-tangent of A/B. The two main differences to atan() is that it will return the right angle depending on the signs of A and B (atan() can only return values betwen -pi/2 and pi/2), and that the return value of pi/2 and -pi/2 are possible.
atanh(A)	Same as atan() but for hyperbolic tangent.
ceil(A)	Ceiling of A. Returns the smallest integer greater than A. Rounds up to the next higher integer.
cos(A)	Cosine of A. Returns the cosine of the angle A, where A is measured in radians.
cosh(A)	Same as cos() but for hyperbolic cosine.
cot(A)	Cotangent of A (equivalent to 1/tan(A)).
csc(A)	Cosecant of A (equivalent to 1/sin(A)).

exp(A)	Exponential of A. Returns the value of e raised to the power A where e is the base of the natural logarithm, i.e. the non-repeating value approximately equal to 2.71828182846.
floor(A)	Floor of A. Returns the largest integer less than A. Rounds down to the next lower integer.
if(A,B,C)	If int(A) differs from 0, the return value of this function is B, else C. Only the parameter which needs to be evaluated is evaluated, the other parameter is skipped; this makes it safe to use eval() in them.
int(A)	Rounds A to the closest integer. 0.5 is rounded to 1.
log(A)	Natural (base e) logarithm of A.
log10(A)	Base 10 logarithm of A.
max(A,B)	If A>B, the result is A, else B.
min(A,B)	If A<B, the result is A, else B.
sec(A)	Secant of A (equivalent to 1/cos(A)).
sin(A)	Sine of A. Returns the sine of the angle A, where A is measured in radians.
sinh(A)	Same as sin() but for hyperbolic sine.
sqrt(A)	Square root of A. Returns the value whose square is A.
tan(A)	Tangent of A. Returns the tangent of the angle A, where A is measured in radians.
tanh(A)	Same as tan() but for hyperbolic tangent.

Examples of function string:

1+2
x-1
-sin(sqrt(x^2+y^2))
*sqrt(XCoord*XCoord + YCoord*YCoord)*

VisIVO Filters

Merge Tables

This operation creates a new table from two or more existing data tables . Up to 100 tables can be merged. Volumes can be merged but they must have the same geometry.

Usage:

VisIVOFilters --op merge [--size HUGE/SMALLEST] [--pad value] [--out filename_out.bin] [--help] [--filelist] table_param.txt

Example:

VisIVOFilters --op merge --out out_table_file.bin --filelist tab_selection_file.txt

the tab_selection_file.txt is a file that contain a list of tables and valid columns. Wildcard "*" means all the columns of the given table.

tab1.bin Col_1
tab1.bin Col_2
tab5.bin Col_x
*tab4.bin **

This command produces a new table (*filename_out.bin* and *filename_out.bin.head*) having columns Col_1 and Col_2 from tab1.bin, Col_x from tab5.bin and all the columns of tab4.bin

Note:
--size SMALLEST default option. Produce a new table having the size of the smallest table. HUGE produces a new table having the size of the greatest table.
--pad Pads the table rows of the smallest table with the given value if HUGE size is used. Default value is 0.
--out Name of the new table. Default name is given.
--filelist (obsolete --file) Input filename containing the table list.

VisIVO Filters

Module

This operation creates a new table (or a new field) computing the module of three fields of the input data table.

Usage:

VisIVOFilters --op module --field parameters [--append] [--outcol colname]
[--out filename_out.bin][--help] [--file] inputFile.bin

Example:

VisIVOFilters --op module --field Vx Vy Vz --outcol Module --append --file inputFile.bin

This command appends a new field named Module to the inputFile.bin file that represents the module of the fields Vx, Vy and Vz sqrt(Vx^2+Vy^2+Vz^2).

Note:
--field Three valid columns name lists used to compute the module.
--append No new table will be created. The original table will have the new field. Default options: a new table with only the new field is produced.
--outcol. Column name of the new field
--out Name of the new table. Default name is given. Ignored if --append is specified.
--file Input table filename

VisIVO Filters

Multi-resolution

This operation creates a new VBT as a random subsample from the input table, with different resolutions.

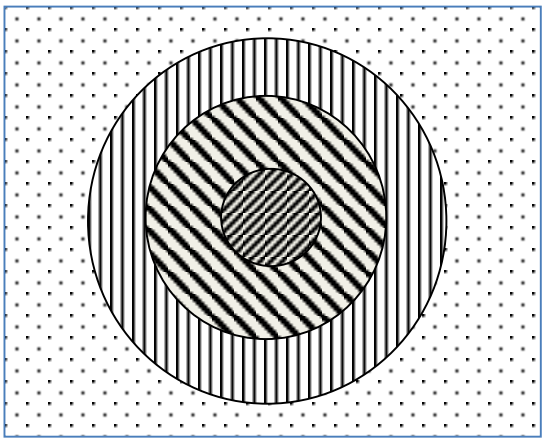
Starting from a fixed position, that represent the center of inner sphere, concentric spheres are considered. Different level of randomization can be given, creating more detail table in the inner sphere and lower detail in the outer regions, or vice versa. The region that is external to the last sphere represents the background.

Usage:

VisIVOFilters --op mres --points x_col y_col z_col [--pos values] [--geometry layer_file.txt] [--background vaule] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op mres --points X Y Z --pos 10.0 10.0 10.0 --geometry layer.txt --background 0.0001 --out pos_layer.bin --file pos.bin



produces a new table (*pos_layer.bin* and *pos_layer.bin.head*).
The pos parameter specify the center of the spheres. The geometry file (*layer.txt*), in this example, has the following values:

5.0 1.0
10.0 0.1
30.0 0.01

5.0 is the radius of the inner sphere. The 1.0 is the percentage of randomization inside the inner sphere: all points that are inside the inner sphere will be reported in the output VBT.
10.0 is the radius of the second sphere. The value 0.1 is the percentage of randomization inside this sphere: only 10% of points that are inside this sphere will be reported in the output VBT.
30.0 is the radius of the last sphere. The value 0.01 is the percentage of randomization inside this sphere: only 1% of points that are inside this sphere will be reported in the output VBT.
background 0.001: only 1 $^{0}/_{00}$ of points that are outside the last sphere, will be reported in the output VBT.

Note:
--points columns to be assumed for points coordinates.
--pos camera point coordinates. Default value is the center of the domain.
--geometry file that contains a radius and a randomization value: 1.0 all values included. 0.1 means 1 per cent of value included in the layer. Each row of this file determine a layer. A default geometry is created with three spheres and different levels of randomization, depending on the input dataset.
--background a randomization value for points outside the geometry. Default value is fixed to have a maximum 100,000 points from input VBT.
--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Point Distribute

This operation creates a volume using a field distribution, CIC (default) or NGP or TSC algorithm, on a regular mesh. The filter produces (by default) a density field: the field is distributed and divided for the cell volume.

Usage:

VisIVOFilters --op pointdistribute --resolution x_res y_res z_res --points x_col y_col z_col [--field column_names] [--constant value] [--nodensity] [--tsc] [--ngp] [--out filename_out.bin] [--gridOrigin x0 y0 z0] [--box length] [--gridSpacing sg0 sg1 sg2] [--periodic] [--avg] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op pointdistribute --resolution 16 16 16 --points X Y Z --field Mass Temperature --out filename_out.bin --file inputFile.bin

It produces a new volume table (*filename_out.bin* and *filename_out.bin.head*) on a *16x16x16* mesh using x y and z as point coordinates.
The filter distributes the *Mass* and the *Temperature* elements of points *X Y Z*. Only the *Mass* and *Temperature* fields will be reported in the new table, as a density field.

Note:
--resolution 3D mesh size.
--points Columns to be assumed for points coordinates.
--field valid columns name list to be distributed in the grid.
--constant Assign a constant value to all points, to be distributed in the grid. Ignored if field option is given. Default value is a 1.0 for all points.
--nodensity Overrides the default behaviour. The field distribution is not divided for the cell volume.
--avg Distributes the first field on the volume grid and computes the arithmetic average value on each cell of the first field. The output volume table will have three fields. For each cell there will be:the number of total elements in the cell NumberOfElements, the sum of total field value fieldSum, and the arithmetic average value fieldAvg. Only the ngp algorithm will be applied.
`--tsc` The TSC algorithm is adopted.
`--ngp` The NGP algorithm is adopted.
--out Name of the new table. Default name is given.
--gridOrigin. It specifies the coordinates of the lower left corner of the grid. Default values are assumed from the box of inputFile.bin
--box. It specifies the length of a box. Default values is assumed from the box of inputFile.bin if the gridSpacing option is not given
--gridSpacing. It specifies the length of each cell dimension in the arbitrary unit. This parameter is ignored if the box option is given. Default values are assumed from the box of the inputFile.bin
--periodic applies a periodical boundary condition
--file Input table filename.

VisIVO Filters

Poca

This operation creates a set of points and a volume starting from a VBT file that contains x,y on four logical planes (muon track reconstruction). The z value on each plane is given as parametric value.

Usage:

VisIVOFilters --op poca [--resolution x_res y_res z_res] [--dimvox voxel_size] [--trackplanedist distance] [--innerdist distance] [--outpoints points.bin] [--outvol vol.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op poca --resolution 600 300 300 --dimvox 10 --outpoins points --outvol vol.bin --file inputFile.bin

Produces a points file VBT with scatter points inside the volume of reference a 60x30x30 and a multi-volume.

The input file is a VBT with 10 columns that represent: Event number, X_A Y_B X_C Y_D X_E Y_F X_G Y_H (8 values coordinates in cm at the planes of the system), Pulse energy in GeV/C The filter gives the track Id and the Energy of each points *X Y Z*. This is the output of the Importer "muportal"
It produces two tables (*points.bin* and *vol.bin*) containing:

The points of the POCA algorithm and angles: 7 fields : event number, X, Y and Z, theta, theta square, energy
A multi-volume (7 volumes) with cell resolution/dimvox on each directiom. Each voxel contatins quantities of all scattering points inside the voxel: sum of the theta (scattering angle), sum of theta square, theta square average, sigma, error and number of scatter points.

Note:
--resolution 3D mesh size in cm. Default value 600 300 300
--dimvox Cubic voxel dimension in cm. Default value 10
--trackplanedist Distance in cm between planes 1 - 2 and planes 3 - 4. Default value 100
--innerdist Distance in cm between planes 3 - 4. Default value 300
--outpoints Name of the new table containing poca points
--outvol Name of the new table containing volumes
--file Input table filename.

VisIVO Filters

Point Property

This operation assigns a property to each data point on the table. The operation performs the following:
1) It creates a temporary volume using a field distribution (CIC algorithm) on a regular mesh. The temporary file will have the filename given in --out option + "_tempPDOp.bin" or "./_tempPDOp.bin", and it is automatically cleaned by the operation itself.
2) It computes, with the same CIC algorithm, the property for each data point, considering the cells where the point is spread on the volume;
3) It saves the property in a new table or adds the field to the original input table.

This operation cannot be applied to volumes.

Usage:

VisIVOFilters --op pointproperty --resolution x_res y_res z_res --points x_col y_col z_col [--field column_name] [--constant value] [--append] [--out filename_out.bin] [--outcol col_name] [--periodic] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op pointproperty --resolution 16 16 16 --points X Y Z --field Mass --append --outcol distribute --file inputFile.bin

The filter distributes the *Mass* field of points *X Y Z and it produces a temporary volume*. Then it calculates a new field that for each X Y Z point, representing the weight value of the nearest mesh-cells where the point is distributed using the same CIC algorithm.

Note:
--resolution 3D mesh size.
--points Columns to be assumed for points coordinates.
--field valid columns name list to be distributed in the grid.
--constant Assign a constant value to all points, to be distributed in the grid. Ignored if field option is given. Default value is a 1.0 for all points.
--append the input table will contain the new field.
--out Name of the new table. Default name is given (ignored if –append is given).
--outcol New field column name.
--periodic applies a periodical boundary condition
--file Input table filename.

VisIVO Filters

Randomizer

This operation creates a sub-table as a random subsample from the input table.

Usage:

VisIVOFilters --op randomizer --perc percentage [--field parameters] [--iseed iseed] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op randomizer --perc 10.0 --iseed 1 --out filename_out.bin --file inputFile.bin

produces a new table (*filename_out.bin* and *filename_out.bin.head*) of 10 percent size of the original inputFile. Values are extracted in a random sequence.

Note:
--perc Percentage (from 0.0 to 95.0) of the input file obtained in the output file. Note: only the first decimal place is considered.

--field (obsolete --list) Valid columns names of the input table. Default: all columns are included.
--iseed Specifies the seed for the random generation. Default value 0.
--out Output table filename. Default name is given.
--file Input table filename

VisIVO Filters

Select Columns

This operation creates e new table using (or excluding) one or more fields of a data table. The default case produces the output table including only listed fields

Usage:

VisIVOFilters --op selcolumns --field parameters [--delete][--out filename_out.bin] [--help] [--file] inputFile.bin

Example

VisIVOFilters --op selcolumns --field X Y --out filename_out.bin --file inputFile.bin

produces a new table (*filename_out.bin* and *filename_out.bin.head*) with only *X* and *Y* columns of inputFile..

Note:
--field (obsolete --list) Valid columns names of the input table.
--delete Produces the output table excluding only field listed in the --field option.
--out Output table filename. Default name is given.
--file Input table filename.

VisIVO Filters

Select Rows

This operation creates a new table setting limits on one or more fields of a data table. Optionally it creates a list of elements satisfying the requested condition.

Usage:

VisIVOFilters --op selfield --limits filename_limits [--operator AND/OR] [--outlist list_filename] [--format uns/int/ascii] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op selfield --limits limitsfile.txt --operator AND --out filename_out.bin --file inputFile.bin

The limitsfile.txt file must have the following structure. A valid column name and an interval indicating the extraction limits:

X 20.0 30.0
Y 10.0 20.0
Z 0.0 10.0

The command produces a new table (*filename_out.bin* and *filename_out.bin.head*) that contains all the data points of the inputFile table (all columns will be reported) where *X*=[20.0,30.0] AND *Y*=[10.0,20.0] AND *Z*=[0.0,10.0] The **unlimited** word can be used to indicate the infinite value.

Note:
--limits A file that has three columns: a valid column name and an interval indicating the extraction limits.
--operator Limits on all fields listed in --limits option file are combined by default with logic AND operator. If this option is given with OR value the field limits are combined with logic OR operator
--out Output table filename. Default name is given.
--outlist Output list filename containing the numer of the element satisfying the requested condition. Default name is given.
--format Data format in the outlist filename. Default value unsigned long long int.
--file Input table filename.

VisIVO Filters

Show Table

This operation produces an ASCII table of the selected field of the first number of rows (or of all rows)

Usage:

VisIVOFilters --op showtable [--field column_name] [--numrows num_of_rows] [--rangerows fromRow toRow] [--width format_width] [--precision format_precision] [--out filename_out.txt] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op showtable --field X Y Z --numrows 100 --out filename_out.txt --file inputFile.bin

produces an ascii file (filename_out.txt) that contains the first 100 data points of the *X,Y* and *Z* fields of the *inputFile* table.

Note:
--field Valid columns names. Default value of ALL columns will be reported.

--numrows Number of Rows in the Ascii output file. Default value is equal to the number of Rows of the input table.
--rangerows Rows range of the inputFile that will be reported in the ascii output file. Default range is equal to all the rows of the input table. It is ignored if numrows is specified.
--width Field width in the ascii output file. Default value is given. Anyway a blank space is left between two consecutive ascii numbers.
--precision Field precision in the ascii output file. Default value is given.
--out Output ascii filename. Default name is given.
--file Input table filename.

VisIVO Filters

Show Volume

This operation writes on an output ascii file, the volume cells and their values that satisfy given limits

Usage:

VisIVOFilters --op showvol [--field column_name] --limits filename_limits [--operator AND/OR] [--numcells value] [--out filename_out.txt] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op showvol --field density mass --limits limitsfile.txt --operator AND --out filename_out.txt --file inputFile.bin

produces an ascii file that lists all cells where limits are satisfied.
The limitsfile.txt file must have the following structure. A valid column name and an interval indicating the extraction limits:

density 20.0 30.0
mass 10.0 20.0

The keyword **unlimited** can be used to indicate the infinite value.
The output ascii file will report on each row the cell (X, Y and Z) and the field value.

Note:
--field Valid columns names. Default value of ALL columns will be reported.
--limits A file that has three columns: a valid column name and an interval that indicate the limits.
--operator Limits on all field listed in the *--limits* option file are combined by default with logic AND operator. If this option is given with OR value the field limits are combined with logic OR operator.
--numcells Set the maximum number of cells that will be reported in the output.
--out Output ascii filename. Default name is given.
--file Input table filename.

VisIVO Filters

Sigma Contours

This operation creates a new table where one or more fields of a data table have values within (or outside) N sigma contours. The filter can be applied on fields that have a Gaussian distribution. For the selected fields, the filter prints in the stdout the average and the sigma values of the distributions.

Usage:

VisIVOFilters --op sigmacontours [--nsigma nOfSigma] [--field columns_list] [--allcolumns] [--out filename_out.bin] [--help] [--file] filename.bin

Example:

VisIVOFilters --op sigmacontours --nsigma 2 --field F1 F2 F5 --allcolumns --out ncontours.bin --file example.bin

The command produces a new table where columns F1, F2 and F5 have (all of them) values included in 2 sigma contours. The option *--allcolumns* creates a new table with all the columns of the input table, otherwise only F1, F2 and F5 will be reported in the output table. The command also prints in the stdout the average values and the sigma values for F1, F2 and F5 columns.

--nsigma Number of sigma used in the variable selection. Default value: 1 sigma countour
--field List of columns that must have values within N sigma contours. Default value: all the columns in the table"<<std::endl;
--exclude. Values outside N sigma contours will be reported.
--allcolumns. All columns of the input tables will be saved in the output table. But only the corresponding rows for the *--field* selected columns are reported
--out Name of the new table. Default name is given.
--file Input table filename.

VisIVO Filters

Statistic

This operation produces average, min and max value of field and creates an histogram of fields in the input table

Usage:

VisIVOFilters --op statistic [--list columns_name] [--histogram [bin]] [--range min max] [--out result.txt] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op statistic --list X Y --histogram 1000 --range 10.0 100.0 --out result.txt --file inputFile.bin

produces min,max and average values printed in stdout. The command also produces a result.txt file that gives the histogram values of X and Y in the range 10.0,100.0 with 1000 bins.

Note:
--list A valid list of columns name. Default value all columns.
--histogram produces an histogram ascii file with a given number of bins. If the bins number is not specified, the default value is fixed to 10% of the total rows of the input table.
--range produces the results only inside the specified interval.
--out Output ascii filename with histogram
--file Input table filename.

An error is given if there are no data in the specified range.

VisIVO Filters

Split Table

This operation divides an existing table into two or more tables, using a field that will be used to divide the table
Usage:

VisIVOFilters --op splittable [--field column] [--volumesplit direction] [--numoftables numberOfTable] [--maxsizetable MaxMbyteSize] [--hugesplit] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op splittable --field X --numoftables 10 --out filename_out --file inputFile.bin

This command will compute the X range (e.g range 0-100) and will produce 10 tables where the X range will be: 0-10, 10-20 ,20-30 and so on.

Note:
--filed A valid column name along which the table will be split. Must be given to split a table.
--volumesplit Direction (1,2 or 3) along which the volume will be split. Must be given to split a volume
--numoftables give the number of tables in which the input table will be split. It must be greather than 1.

--maxsizetable indicates the maximum size of the split table in MegaBytes. VisIVO Filter will compute how many tables will be created. This option is ignored if *--numoftable* option is given.
--hugesplit. Must be given to force the generation of more than 100 tables from the input table, avoiding errors.
--out Output prefix root table filename. A suffix _split_#.bin is added to each generated table, to this prefix. Default name is given.
--file Input table filename.

VisIVO Filters

Swap

This operation produces the Endianism swap: little Endian to big Endian data format and viceversa of a VisIVO Binary Table. It can produce a new swapped table or data can be overwritten.

Usage:

VisIVOFilters --op swap [--override] [--out filename_out.bin] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op swap --override --file inputFile.bin

data are swapped and overwritten

Note:
--override The same input table is swapped. Data are overwritten.
--out Name of the new table. Default name is given. Ignored if --override is specified.
--file Input table filename.

VisIVO Filters

VBT2AHF

This operation creates an Amiga halofinder (v0.0/271) data file format (serila version) from six columns of a VisIVO table, typically a snapshot of a cosmological simulation.
This operation needs some important parameters that must be given in an external parameter file. It must contain the following data (space separated):

np (int, number of elements),

box (double, box size),
mass_frac (double, mass of each element),
vel_fact (float, factor to transform velocity field in Km/sec),
omega0,
lambda0,
omegab,
gamma,
H_frac,
T_initial,
z_initial,
z_current (double, cosmological parameters),
numberOfTimesteps (int),
header (string, header identifier)

Parameter file example:

8000000
70. 2.6773e+09 150. 2.14e-1 7.42e-1 0.044
1.666 7.19e-1 1.e3 50. 0.2
500 32ml_z0.2

More details on this parameter on the Amiga HF documentation.
Note: for AMIGA HF users. No DONT_KIK_ME, EQ_MOT2, ART, ISOLATED, ISOLATED_P, XDOT are included in this filter

NOTE: The filters is available only if the AHF option is used when compiling VisIVO.

Usage:

VisIVOFilters --op vbt2ahf [--field columns_name] --par parfile [--out filename_out] [--help] --file inputFile.bin

Example:

VisIVOFilters --op vbt2ahf --field X Y Z Vx Vy Vz --par myFile.par --out outFile.amg [--help] --file input.bin

produces yhe outFile.amg file from X Y Z Vx Vy Vz fields of input.bin file and considering the parameter file data.

Note:
--field a valid list of columns names. Default names X Y Z Vx Vy Vz are considered
--par A valid input parameter file.
--out is the output amiga data format file. Default name is given.
--file. Input table. Typically a snapshot of a cosmological simulation.

VisIVO Filters

Visual

This operation creates an eventually randomized new table from one or more input tables. All the input tables must have the same number of rows. The new table can be used with VisIVOViewer. The operation cannot be applied to volume tables.

Usage:

VisIVOFilters --op visualop [--size number_of_elements] [--out filename_out.bin] [--help] [--filelist] tab_selection_file.txt

Example:

VisIVOFilters --op visualop --out filename_out.bin --filelist tab_selection_file.txt

The tab_selection_file.txt is a text file that contain a list of tables and valid columns. Wildcard "*" means all columns of the given table.

tab1.bin Col_1
tab5.bin Col_x
*tab4.bin **
tab1.bin Col_2

The command produces a new table (*filename_out.bin* and *filename_out.bin.head*) having columns *Col_1* and *Col_2* from *tab1.bin*, *Col_x* from *tab5.bin* and all the columns of *tab4.bin*. If the row number of the input tables exceeds *8000000* elements, the output file will be limited to *8000000* randomized sampled rows
The column names in the output table will have the suffix *_visual_#* where # represent the number order listed in the txt file. The output VBT will contain the columns of the listed tables in alphabetic order. In the above example, the header of the VBT (if tab4.bin contains two columns A and B and there are 8000000 rows) will be:

float
5
8000000
little
Col_1_visual_1
Col_2_visual_5
A_visual_3
B_visual_4
Col_x_visual_2

Note:
--size Number of max rows in the output table. Default is the minimum between 8000000 rows and the rows of input tables. The input tables must have the same number of rows.
--out Output table filename. Default name is given.
--filelist *(obsolete --file)* Input text file with a list of tables and columns

VisIVO Filters

Write VOTable

This operation produces a VOTable 1.2 with selected field.

Usage:

VisIVOFilters --op wrvotable [--field column_name] [--force] [--out filename_out.xml] [--help] [--file] inputFile.bin

Example:

VisIVOFilters --op wrvotable --field ra dec u --out votable_out.xml --file inputFile.bin

produces a VOTable (votable_out.xml) that contains data of *ra,dec* and *u* fields of the *inputFile* table.

Note:
--field Valid columns names. Default values of ALL columns will be reported.
--force Must be given to force the VOTable creation when the input table has s large number of rows (1 Million).
--out Output ascii filename. Default name is given.
--file Input table filename.