



AVS/Express and VisIt Training

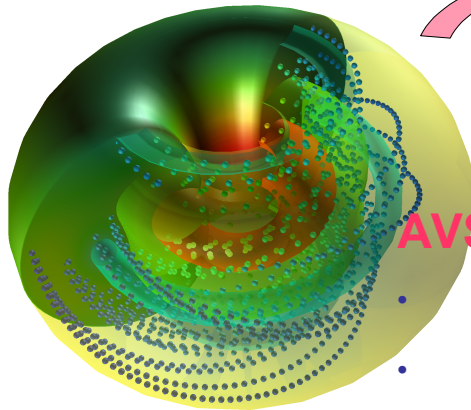
NERSC Users meeting June 2006 - PPPL

Cristina Siegerist
NERSC/LBNL Visualization group
June 13, 2006



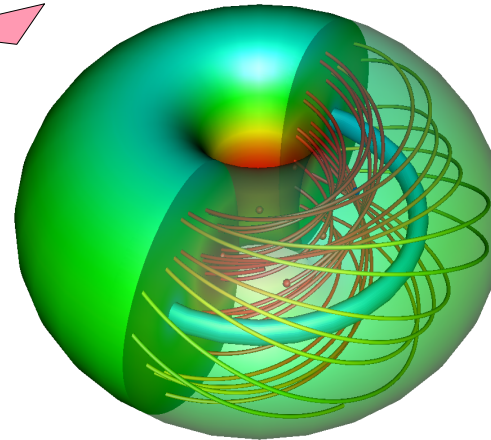
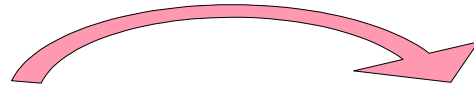


AVS/Express and VisIt Training Overview



AVS/Express

- Introduction
- Network Editor
- Importing Data
- Libraries
- V language
- Module writing



VisIt

- Introduction
- Databases
- Plots
- Operators



Conclusions and
what was left out

How to contact us:

Wes Bethel (group leader), ewbethel@lbl.gov

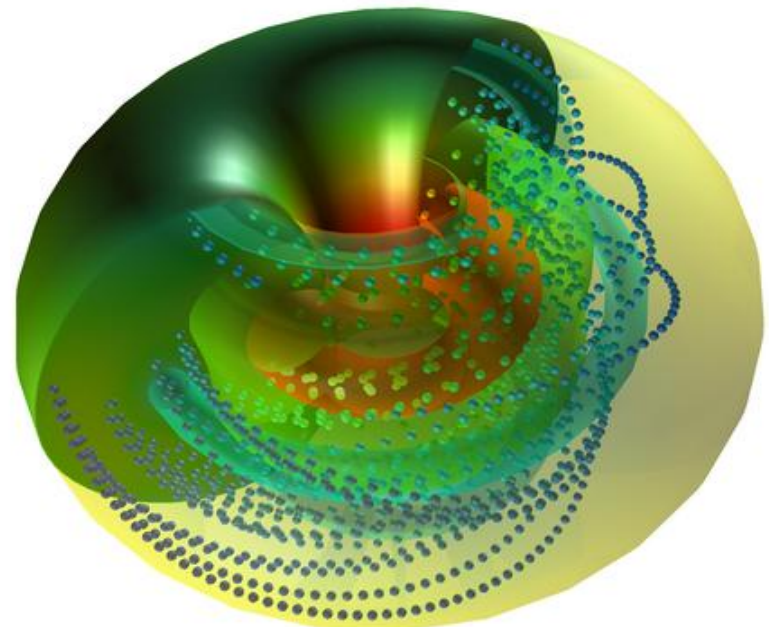
Cristina Siegerist, cesiegerist@lbl.gov

<http://vis.lbl.gov>



What is AVS/Express?

- An object oriented visualization development tool
- Graphical development interface
- Modular, with many built in library modules
- Open and extensible, using V-language and/or C, C++, FORTRAN Api.
- Serial, not distributed.



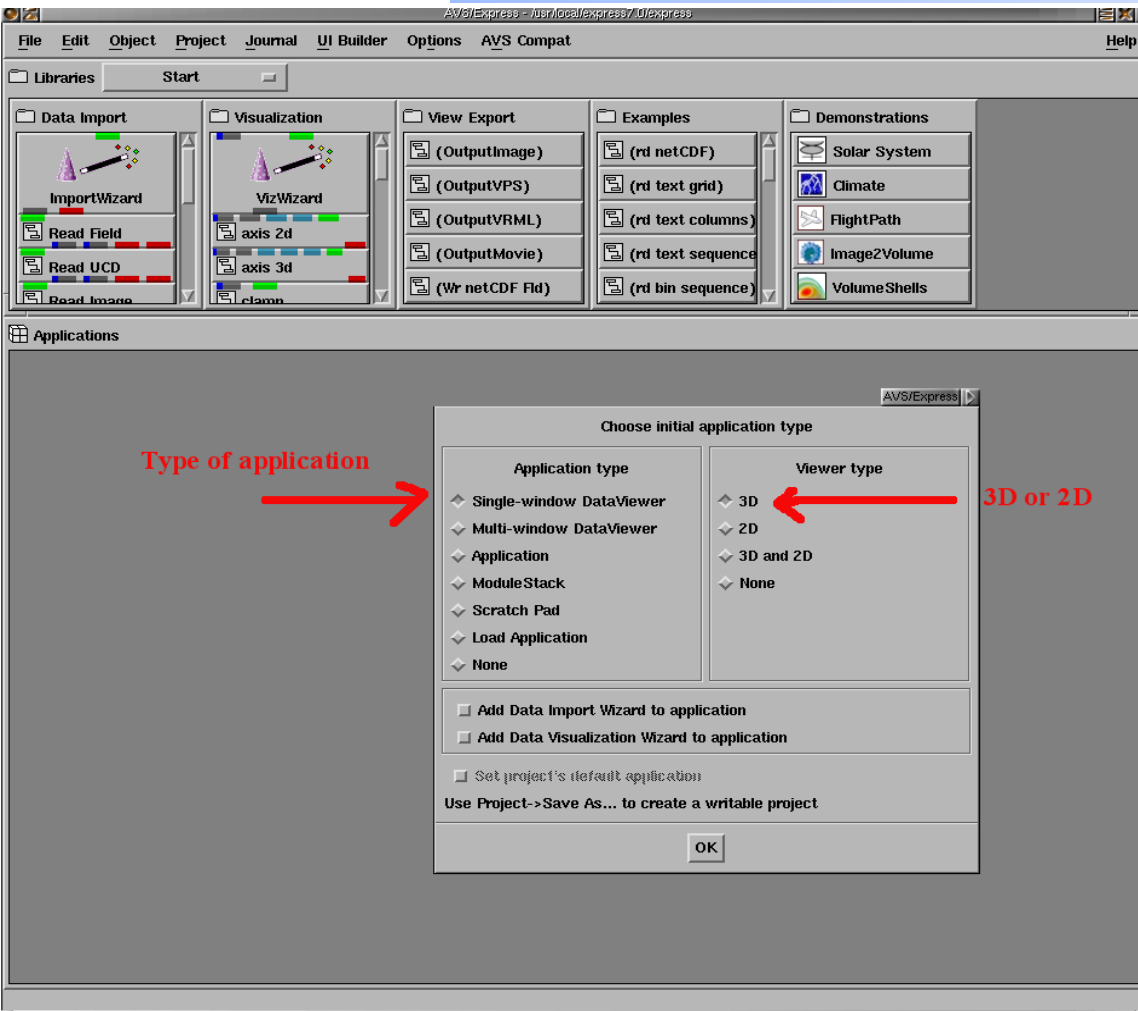


AVS/Express in NERSC

- 250 licenses
- DaVinci (7.0, 7.1), Seaborg(6.3)
- module load avs-express
- %express
- Licenses can be checked out from the license server
<http://www.nersc.gov/nusers/services/licenses/>



Network Editor



Type of application

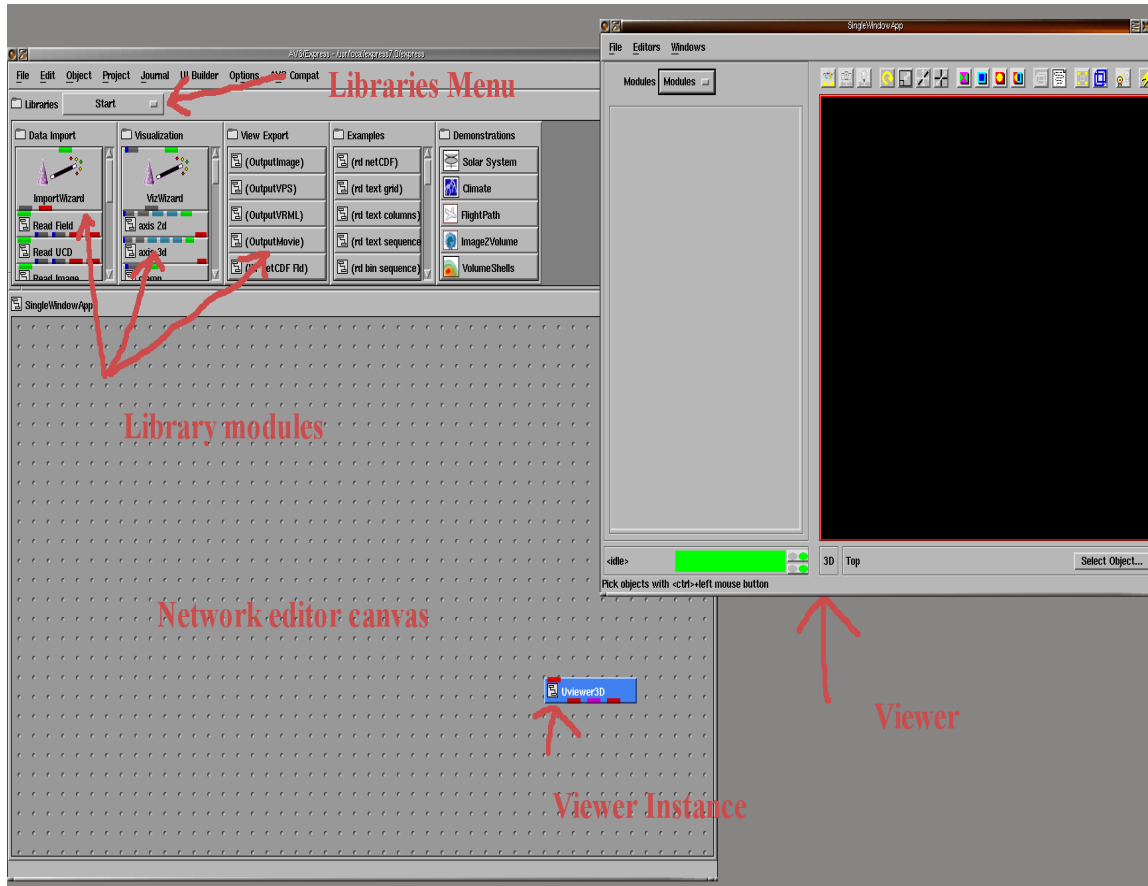
- Single-window
- Multi-window
- Application
- ...

and dimensions.

Choose the default:
Single-window, 3D.



Network Editor

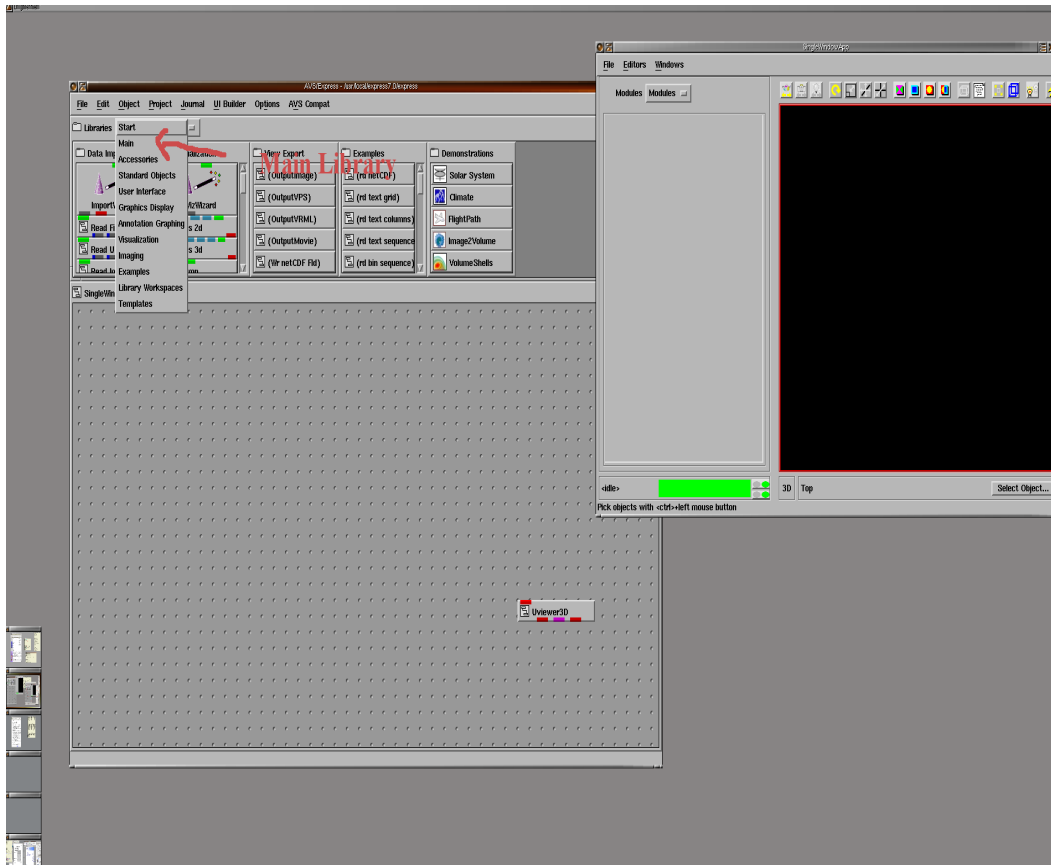


Selecting single window app and 3d you will see:

- 3D Display
- 3D Viewer instance
- Built-in Library modules



Network Editor



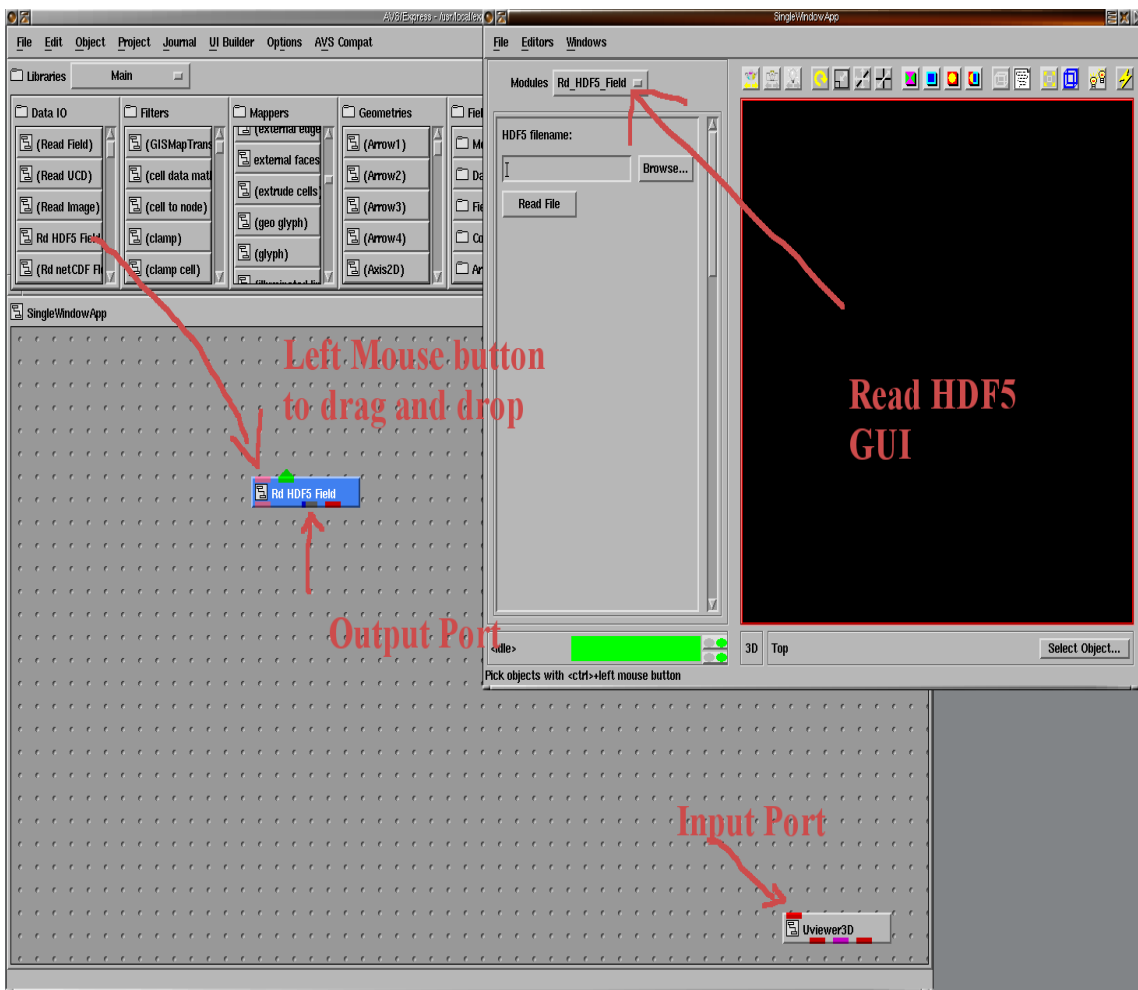
Main Library:

- Data IO**
- Filters**
- Mappers**
- Geometries**
- Field Mappers**
- Viewers**

Network Editor

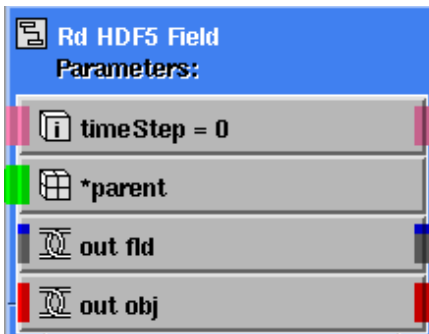
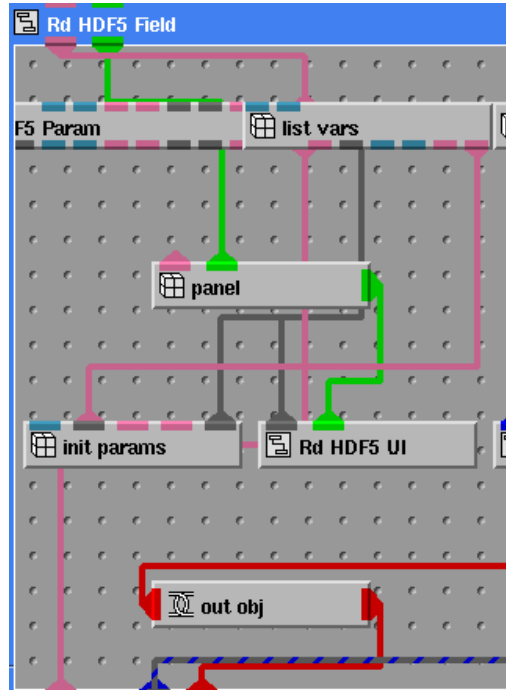
Left mouse to drag and drop instances of modules onto the canvas. Choose “Rd_HDF5_Field” from the Data IO library.

The viewer will show the corresponding gui. From the browser select “./tutorial1.h5”





Network Editor



Left click on a module to select and move.

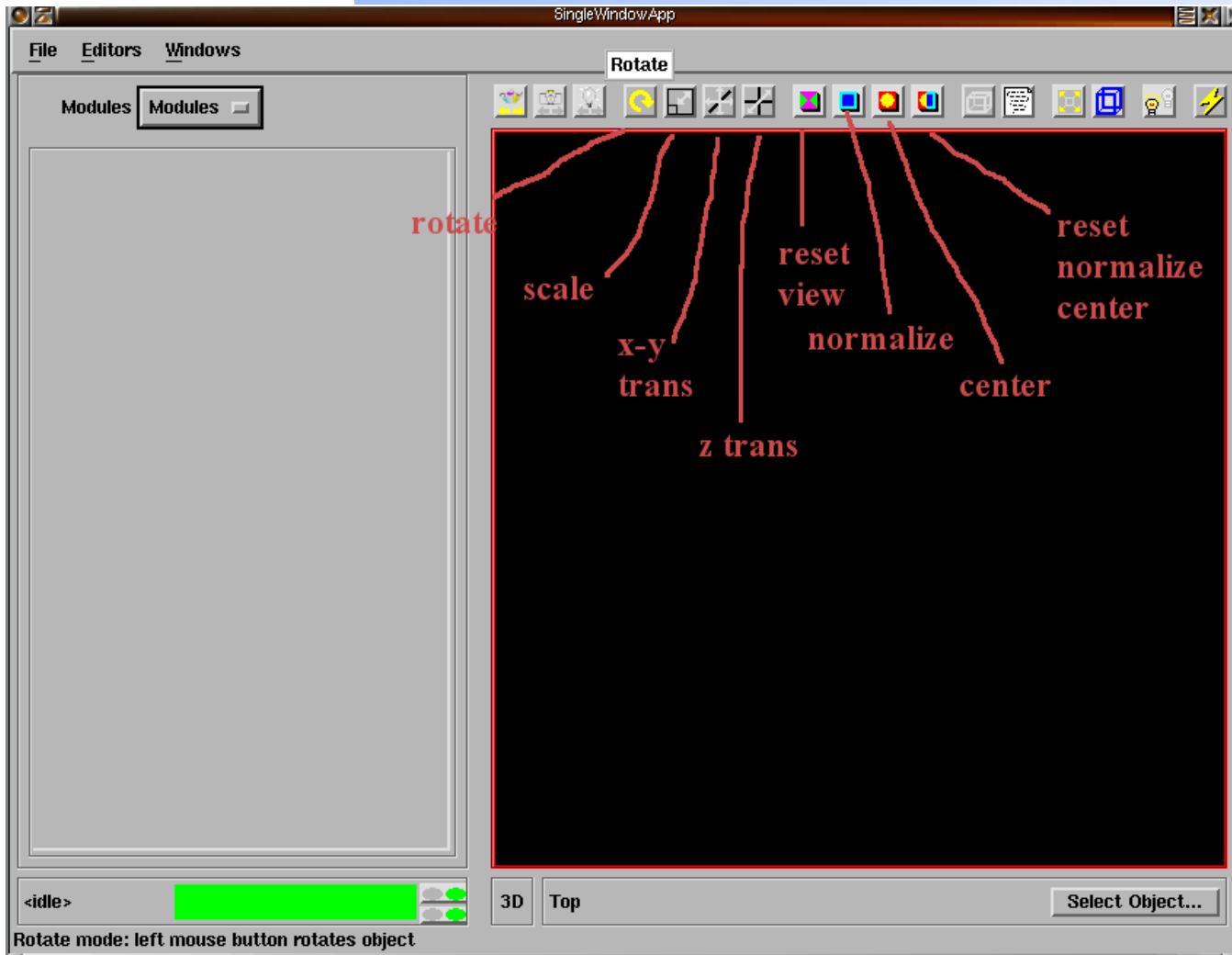
Right click on a module will give a menu of options:

Parameters

Open



Network Editor

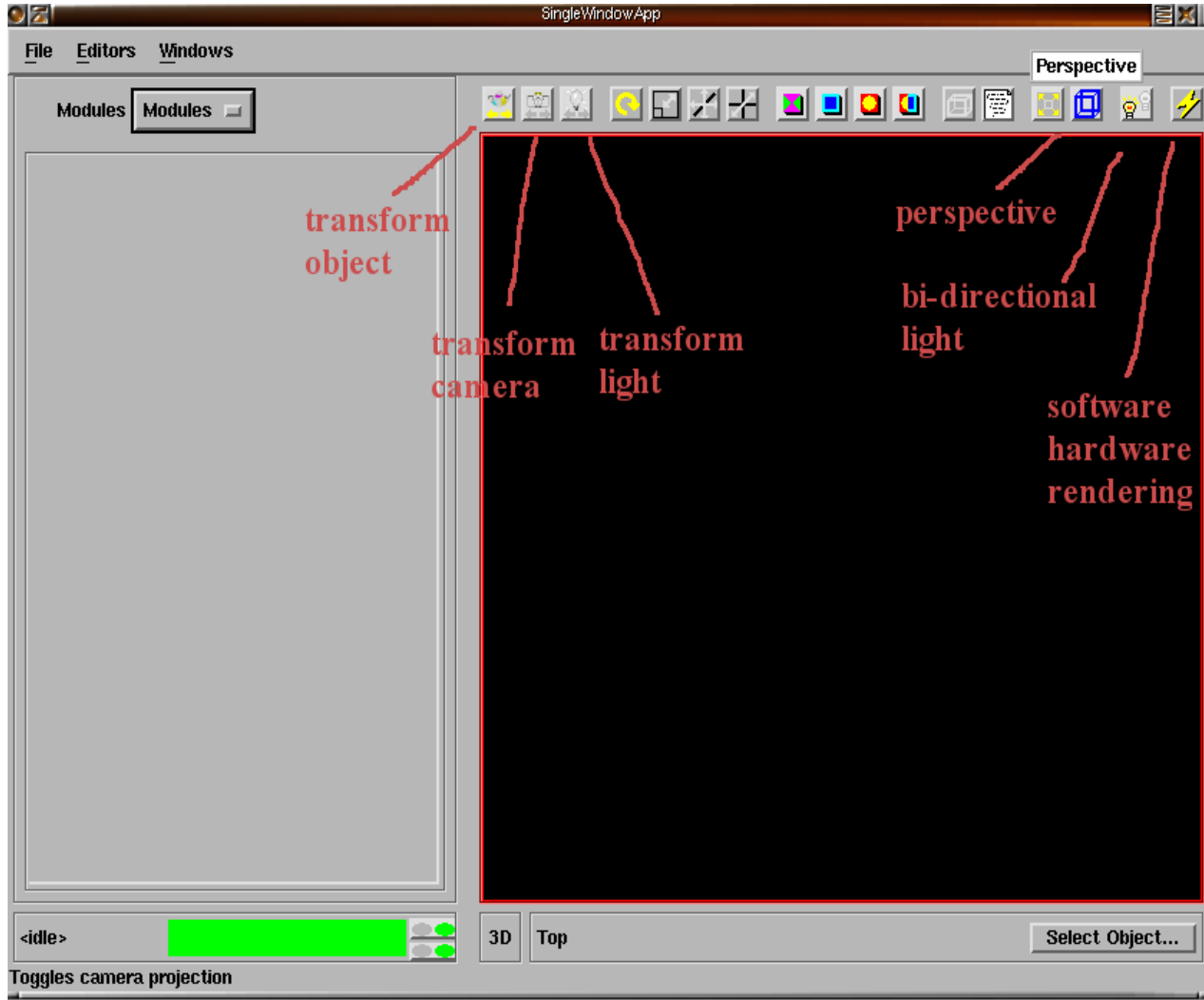


Viewing
controls:
mouse
over the
icons
tells you
what
they are



Network Editor

Viewing
controls
continued





Reading Data

- **Supported Formats**
- **The Express Field**
- **Importing Data into AVS/Express**
- **Writing Custom Readers**



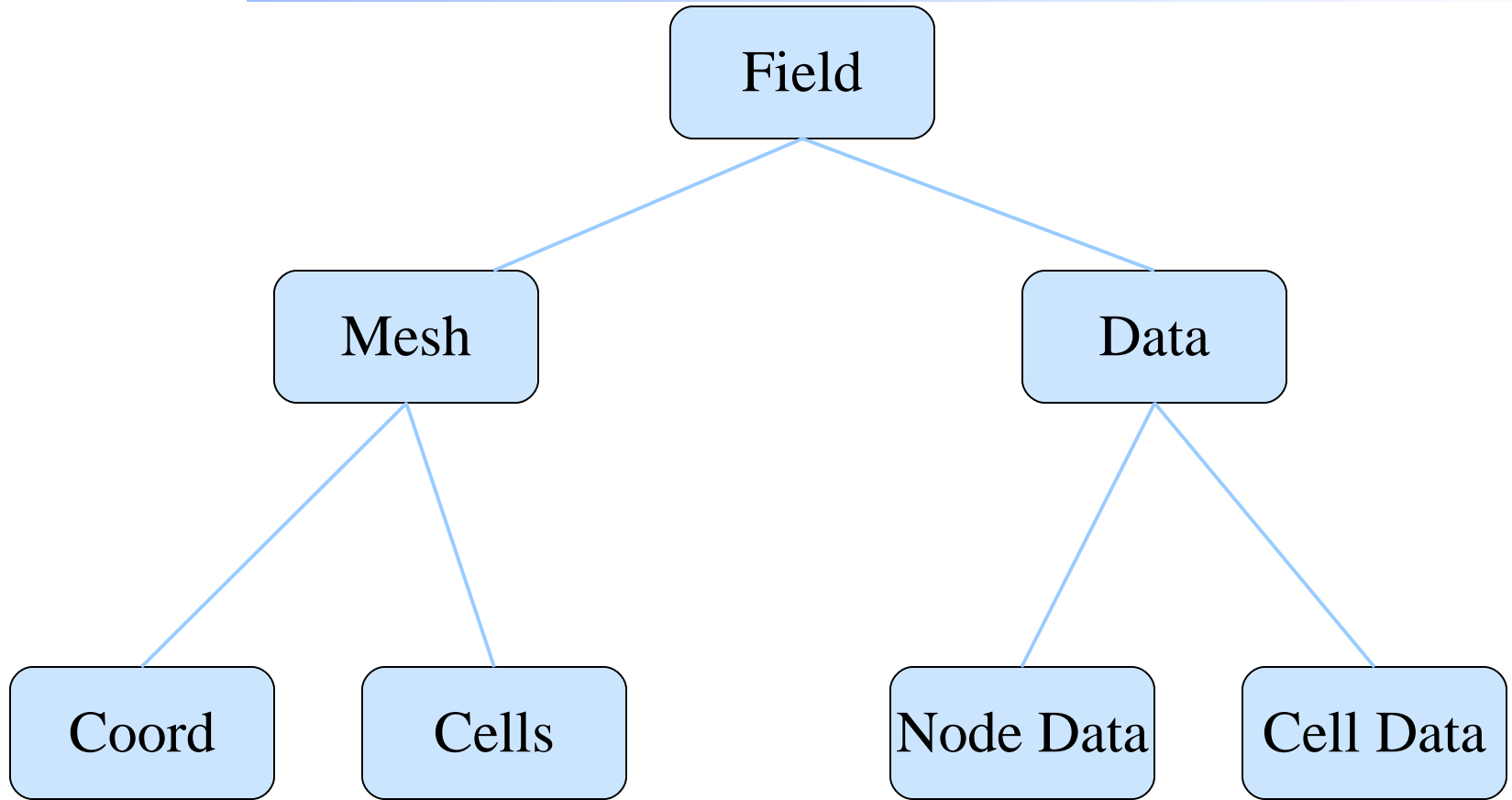
Reading Data

Some Readers (Main, Data IO library)

Read_Field, Read_UCD, Rd_netCDF_Fld,
Rd_HDF5_Field, Read_PLOT3D, Read_DXF,
Rd_Txt_Columns, Rd_Txt_Grid,
Rd_Bin_Sequence, Read_Volumes,
Read_Polygon, Read_Triangle...



The Express Field



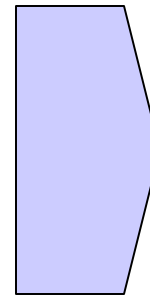


The Express Field

Field Types:

- Structured (or irregular)
- Rectilinear
- Uniform

- Unstructured



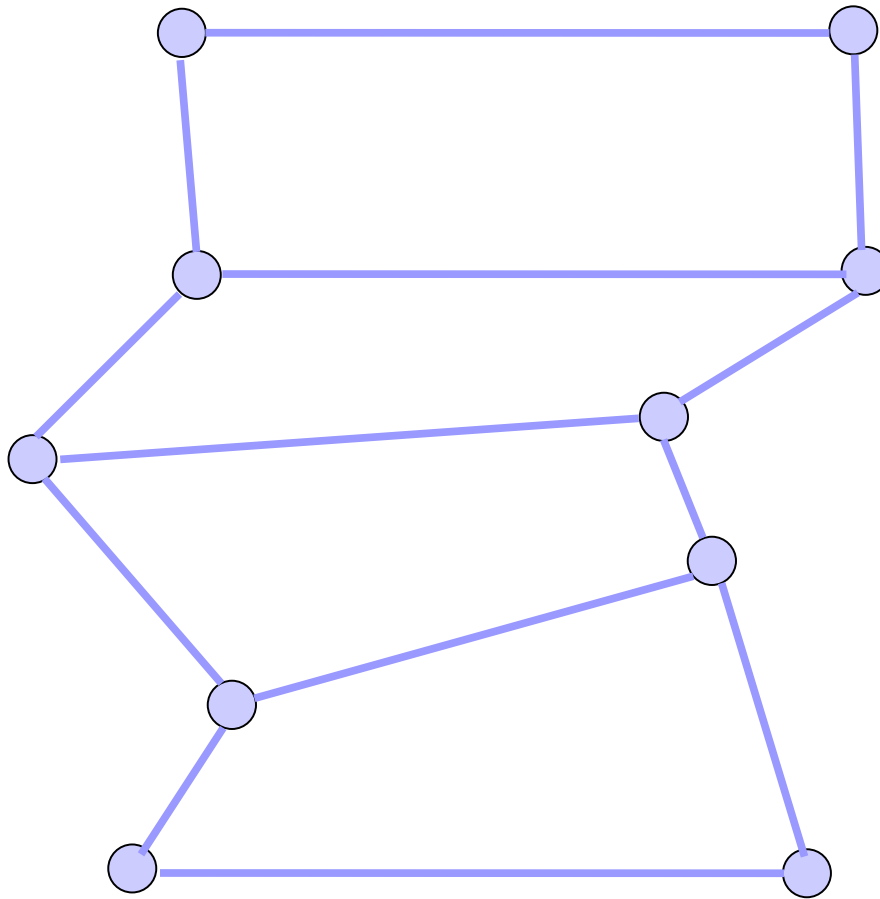
Connectivity is implicit,
can define a header file
to use Read_Field



Connectivity must
be defined.



The Structured Field



Structured Field:
nspace
ndims
dims
coordinates
node data: scalar,
vector, etc.



The Read_Field Module

Header file
for a
structured
field

needs
to be
there

```
# AVS field file
# This is a header file for a structured field
#
#
ndim = 3
dim1 = 40
dim2 = 32
dim3 = 32
nspace = 3
veclen = 5
data = float
field = irregular
label = density x-momentum y-momentum z-momentum stagnation

variable 1 file=./blntfinq.bin filetype=binary skip=28
variable 2 file=./blntfinq.bin filetype=binary skip=163868
variable 3 file=./blntfinq.bin filetype=binary skip=327708
variable 4 file=./blntfinq.bin filetype=binary skip=491548
variable 5 file=./blntfinq.bin filetype=binary skip=655388

coord 1 file=./blntfinx.bin filetype=binary skip=12
coord 2 file=./blntfinx.bin filetype=binary skip=163852
coord 3 file=./blntfinx.bin filetype=binary skip=327692
```



The Read_Field Module

Uniform
Field

```
# AVS field file
# this is a header file for a uniform field
#
#
ndim = 3
dim1 = 64
dim2 = 64
dim3 = 64
nspace = 3
veclen = 1
data = byte
field = uniform

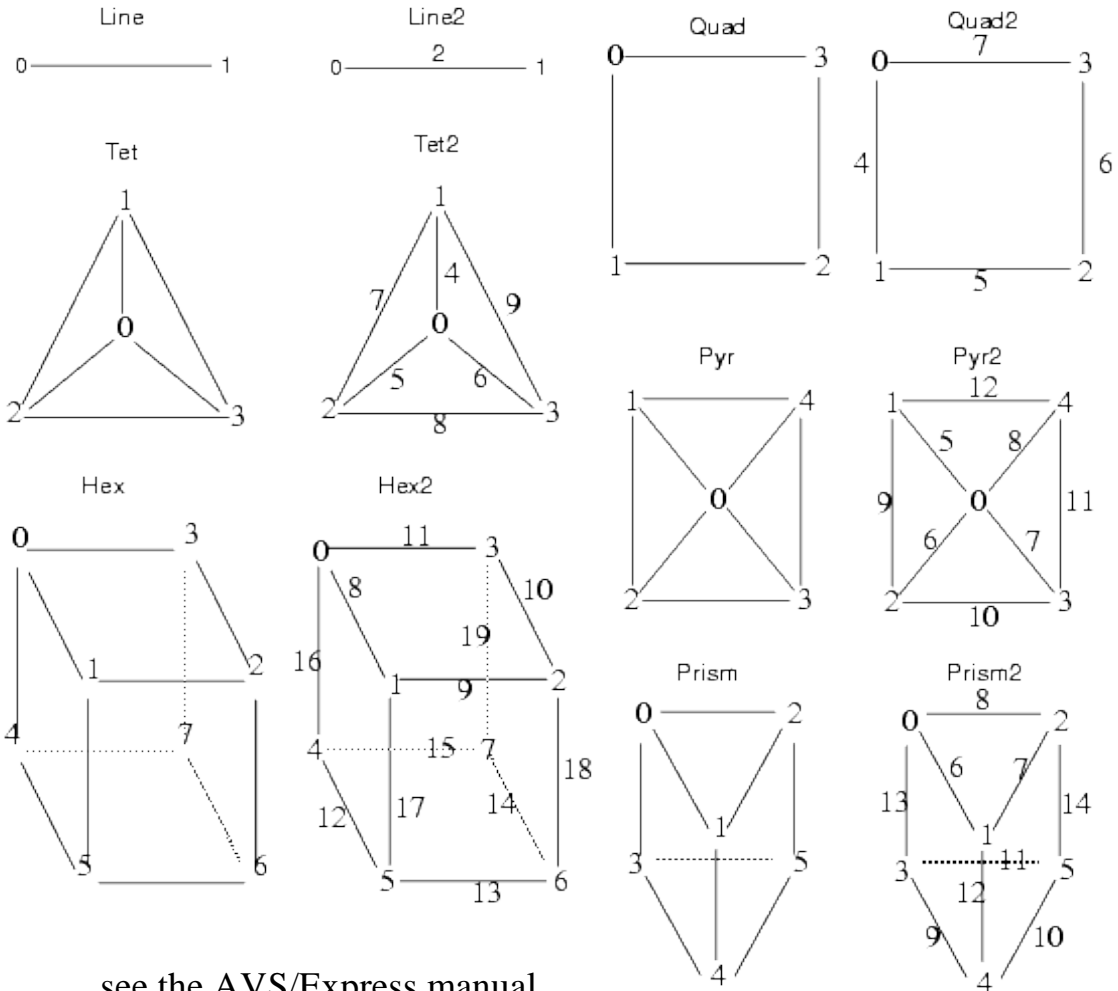
variable 1 file=./hydrogen.dat filetype=binary skip=3
```

Note: FORTRAN unformatted data can be read in by using for example:
variable 1 file=for0004.dat filetype=unformatted skip=32 stride=1



The Unstructured Field

Cell types:



see the AVS/Express manual



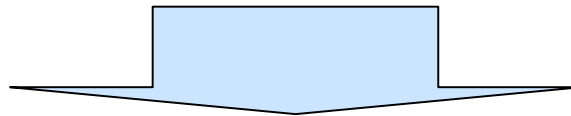
Unstructured Field

- Can be assembled in Express by reading the connectivity array, by writing it in HDF5 format, or by using a custom reader.



Importing Data into Uniform Fields

- **Rd_Text_Grid**: reads in grid-oriented text, often the result of exporting a spreadsheet
- **Rd_Text_Sequence**: reads in a sequence of numbers from an ascii file into a uniform field.
- **Rd_Bin_Sequence**: Reads in a sequence of numbers from a binary file into a uniform field.



Uniform fields



Importing Data into Tables

- **Rd_Text_Columns**: reads an ASCII file containing any number of separated columns into a “table”. It can be converted to the Uniform Field or Scattered (particles) Field types.



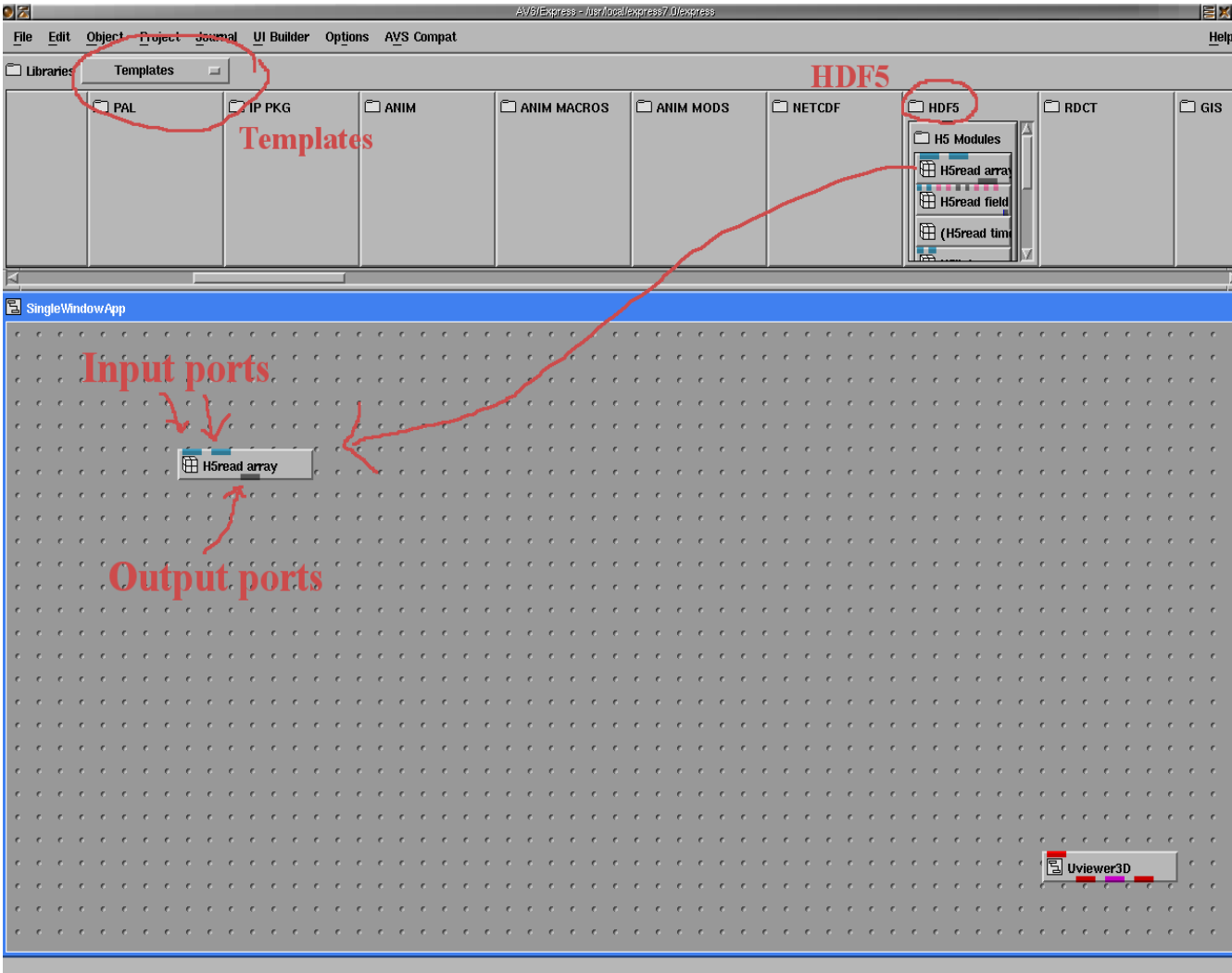
Example Applications

- I - Reading an hdf5 array into a 2D uniform mesh.**
- II - Reading a 3D HDF5 file, making an isosurface, changing the transparency.**
- III – Visualizing a vector field, making streamlines.**



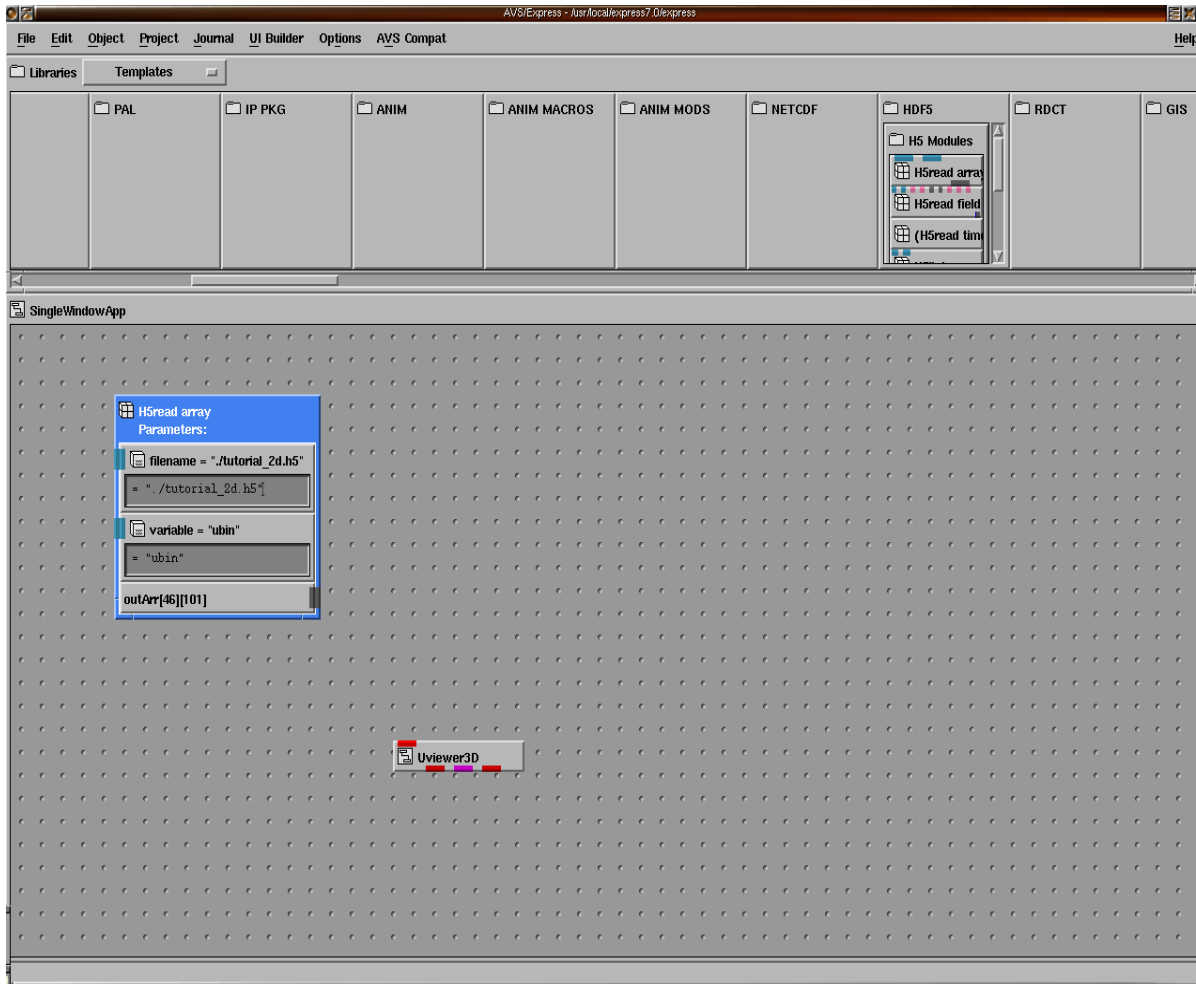
Example Application I

From the
Templates
/HDF5 library
drag and drop:
H5read_array





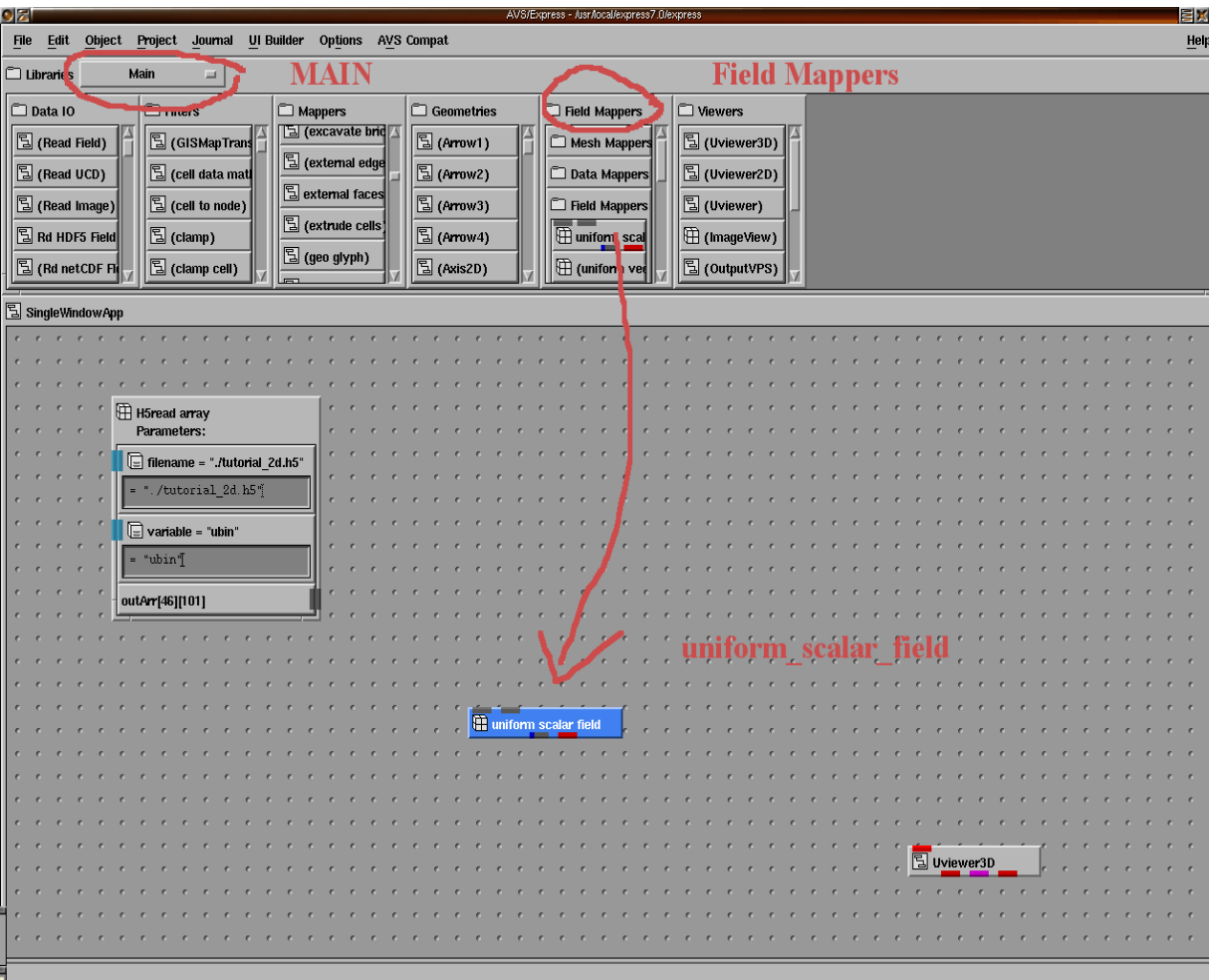
Example Application I



Right click and select open in each of the “string” objects: type “./tutorial_2D.h5” in filename and “ubin” in variable



Example Application I



From Main/Field Mappers drag and drop `uniform_scalar`

Left mouse button on an output port and connect to the corresponding input port of another module.



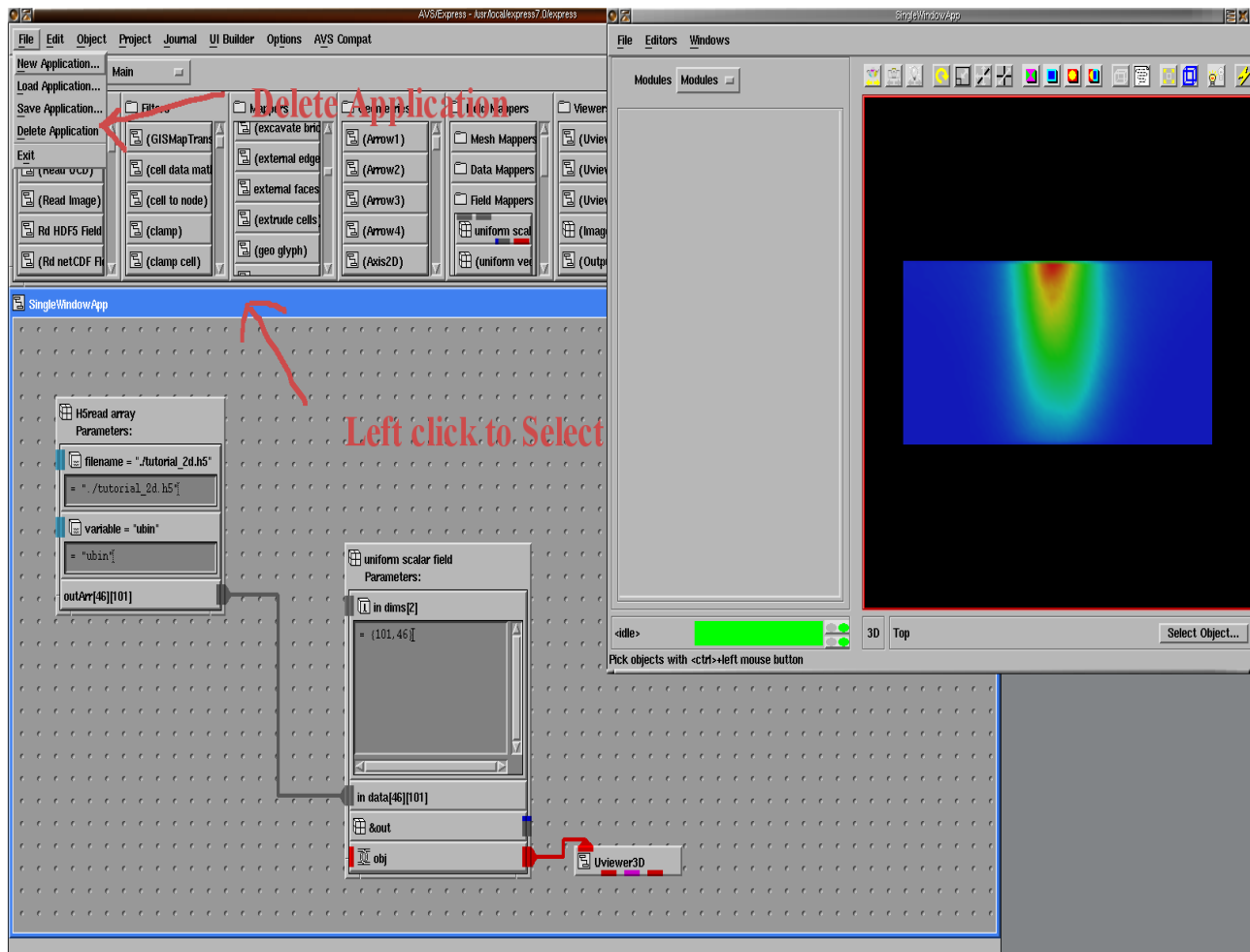
Example Application I

The screenshot displays the AVS/Express software interface. The top-left pane shows a library of modules categorized into Data IO, Filters, Mappers, Geometries, Field Mappers, and Viewers. The main workspace contains a workflow with the following components:

- H5read array**: Parameters include filename = ".tutorial_2d.h5" and variable = "ubin". Output: outArr[46][101].
- uniform scalar field**: Parameters include in dims[2] = {101, 46} (highlighted in red) and in data[46][101]. Output: &out.
- Uviewer3D**: Receives the output from the uniform scalar field module.

The right-hand pane shows a 3D visualization of the data, displaying a color gradient (blue to red) on a rectangular plane. The interface includes a menu bar (File, Edit, Object, Project, Journal, UI Builder, Options, AVS Compat), a toolbar, and a status bar with the text "Pick objects with <ctrl>+left mouse button".

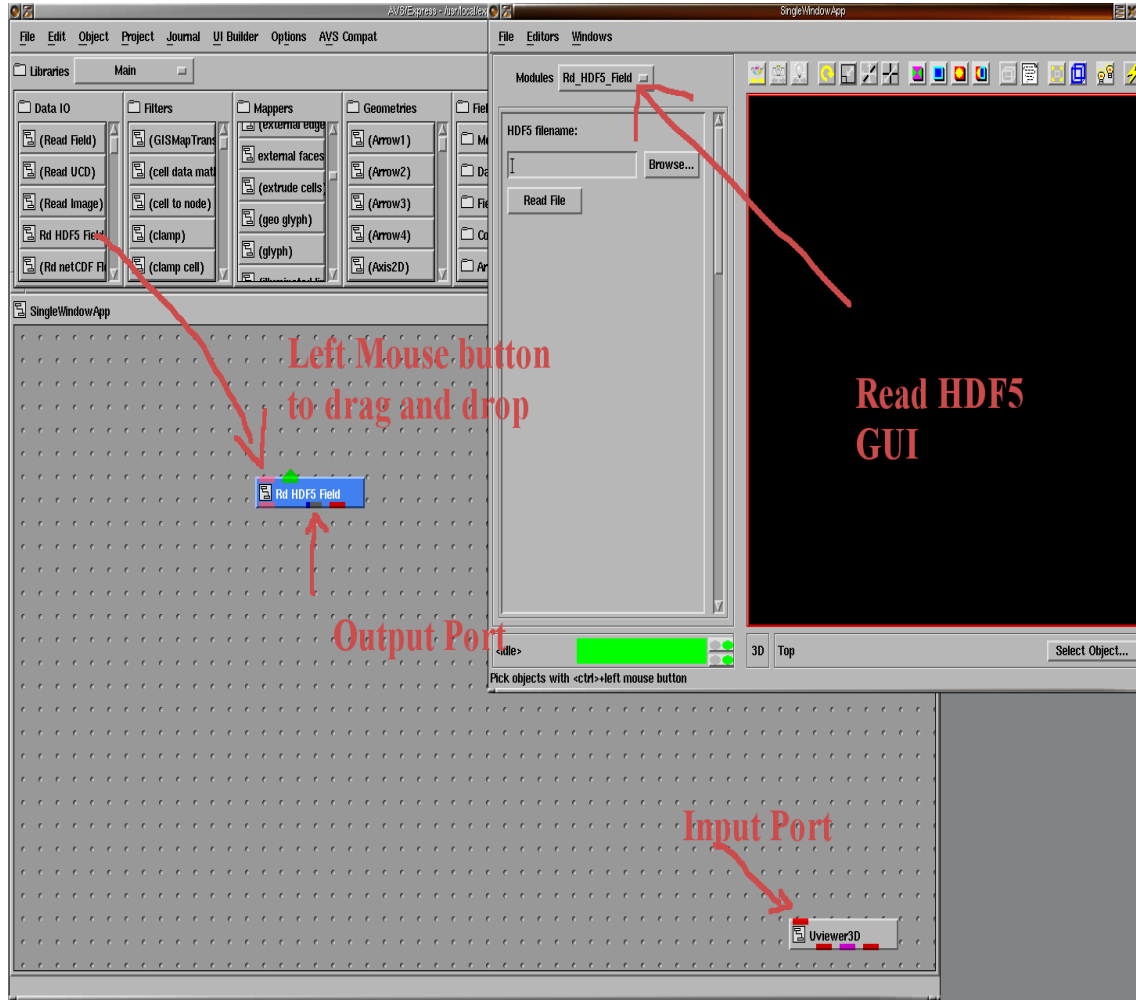
Example Application I



**Delete
the
application:
left click on
SingleWinApp
choose
Delete
Application
from the main
File menu.**



Example Application II

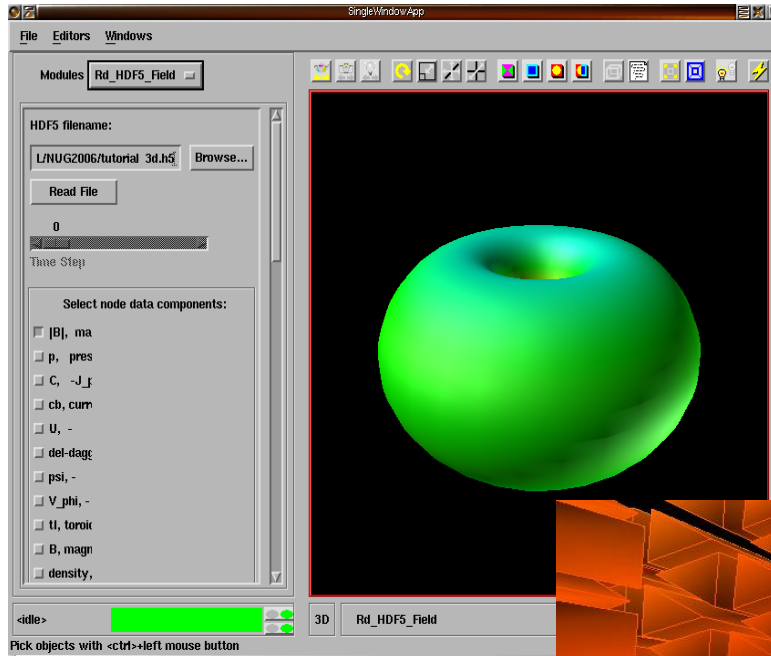


From
MAIN/Data IO
drag and drop
Rd_HDF5_Field

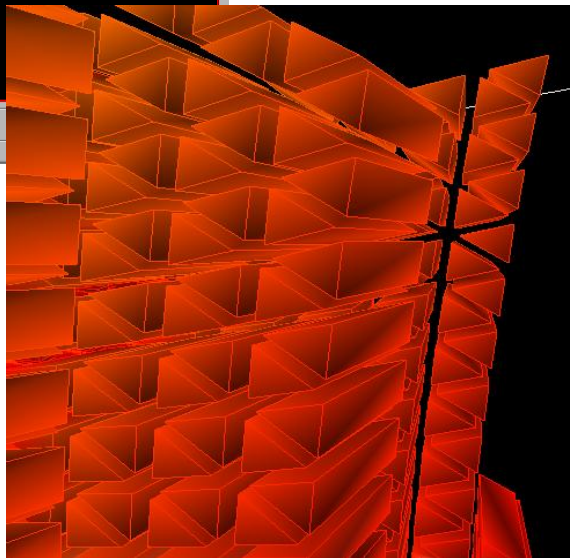
Read
tutorial_3d.h5
from the GUI



Example Application II

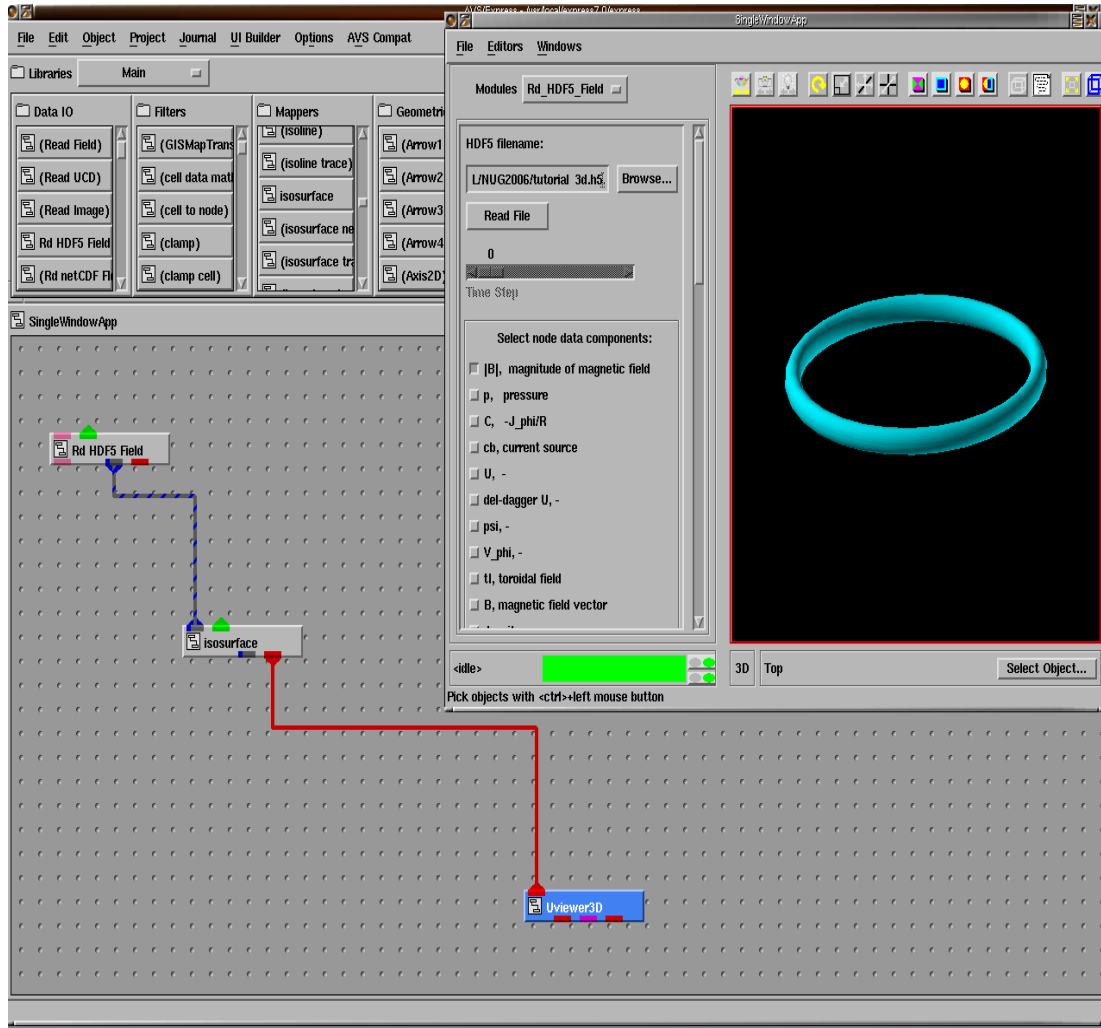


Unstructured data:
mesh made with
prisms and several
node data variables,
both scalar and
vector.





Example Application II

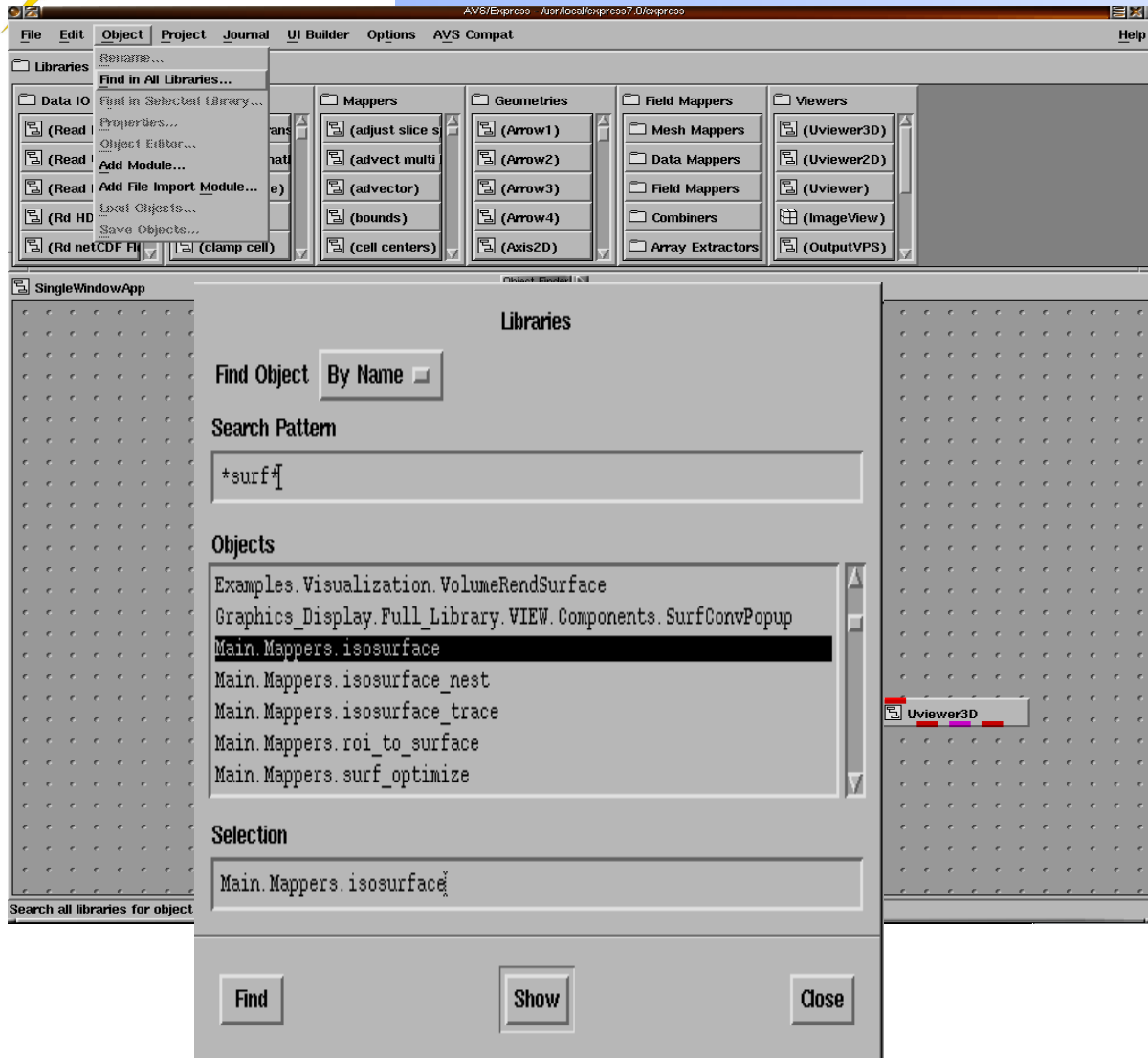


**Find isosurface
(Main/Mappers).**

**Left mouse button
on an output port
and connect to
the corresponding
input port of
another module.**

**Delete a connection:
right click on the
connection and
select delete.**

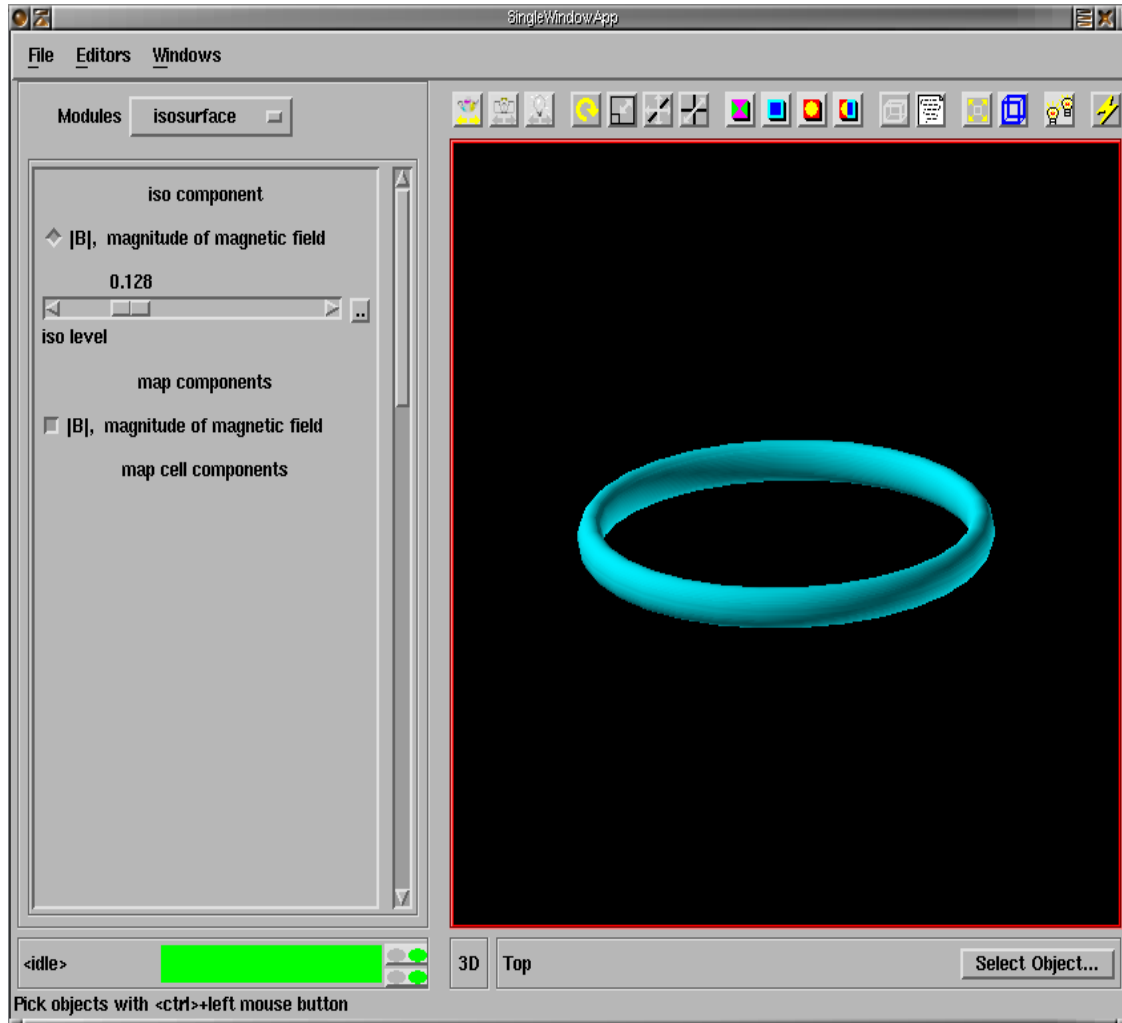
Example Application II



Tip: under the Object menu there is a “Find in All Libraries...” option. Search for *surf* and all the modules surf related will appear. Show will show the module in the library.



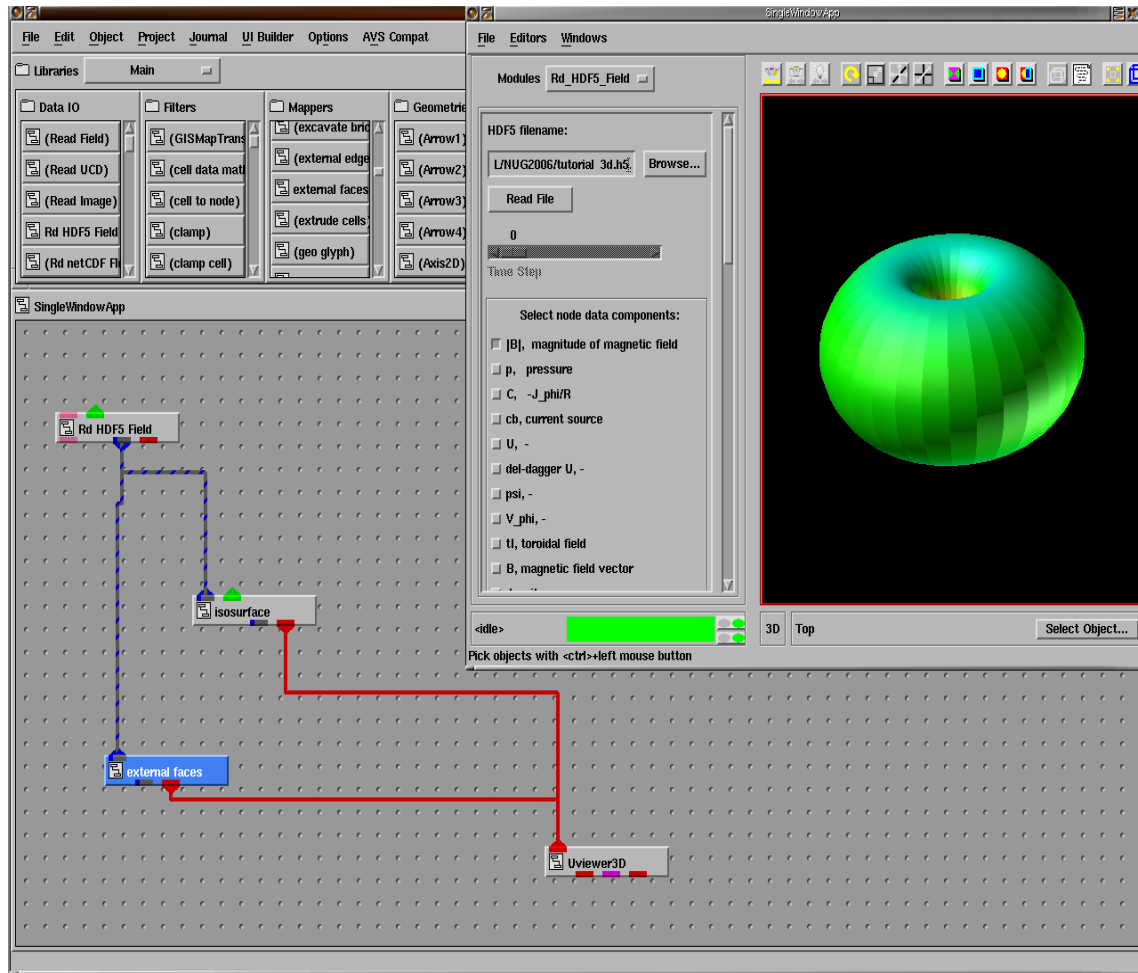
Example Application II



Example Application:
Change the isosurface value from the isosurface module GUI.

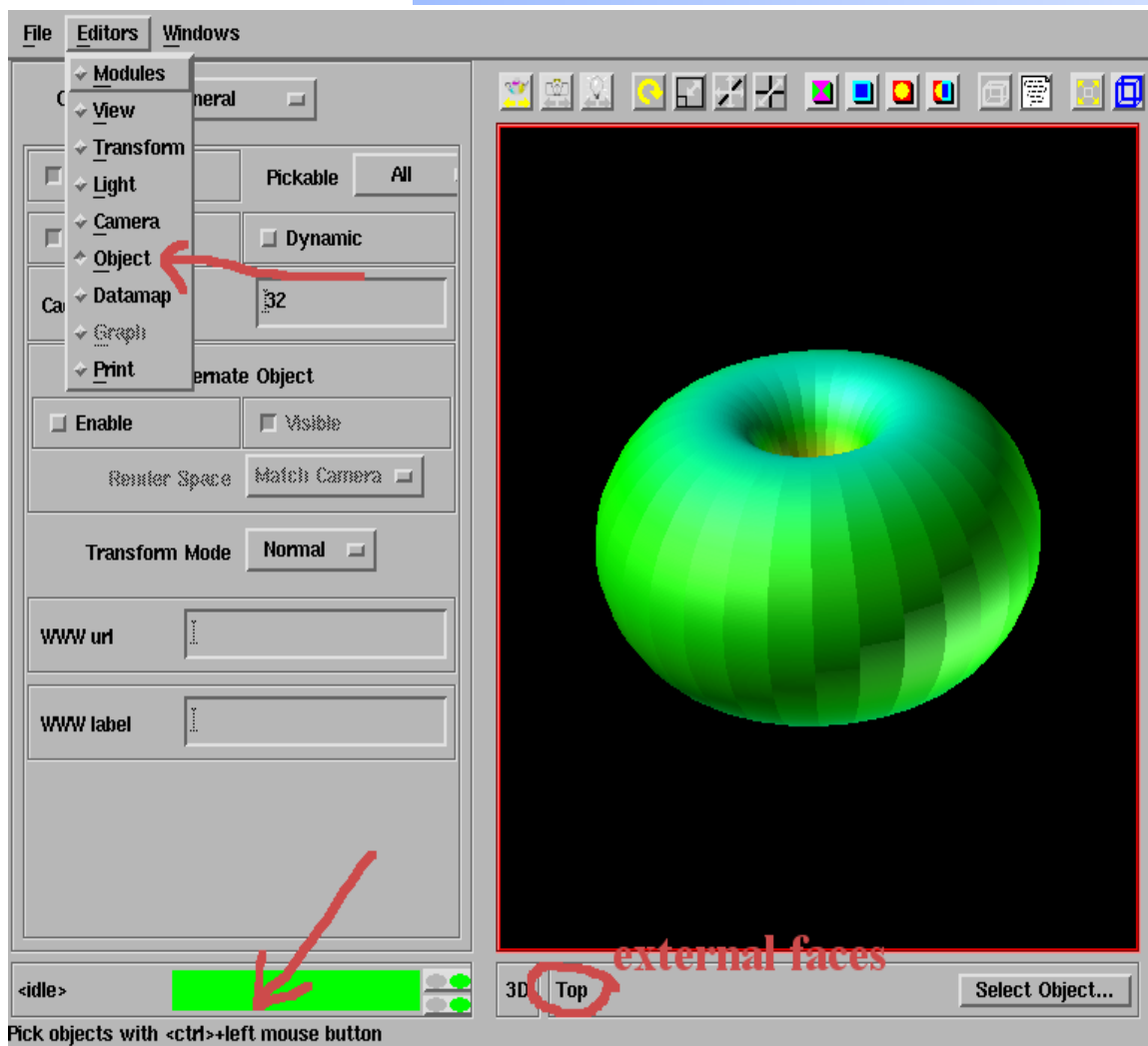


Example Application II



Find the
Main/
external_faces
module

Example Application II



Changing the
transparency:

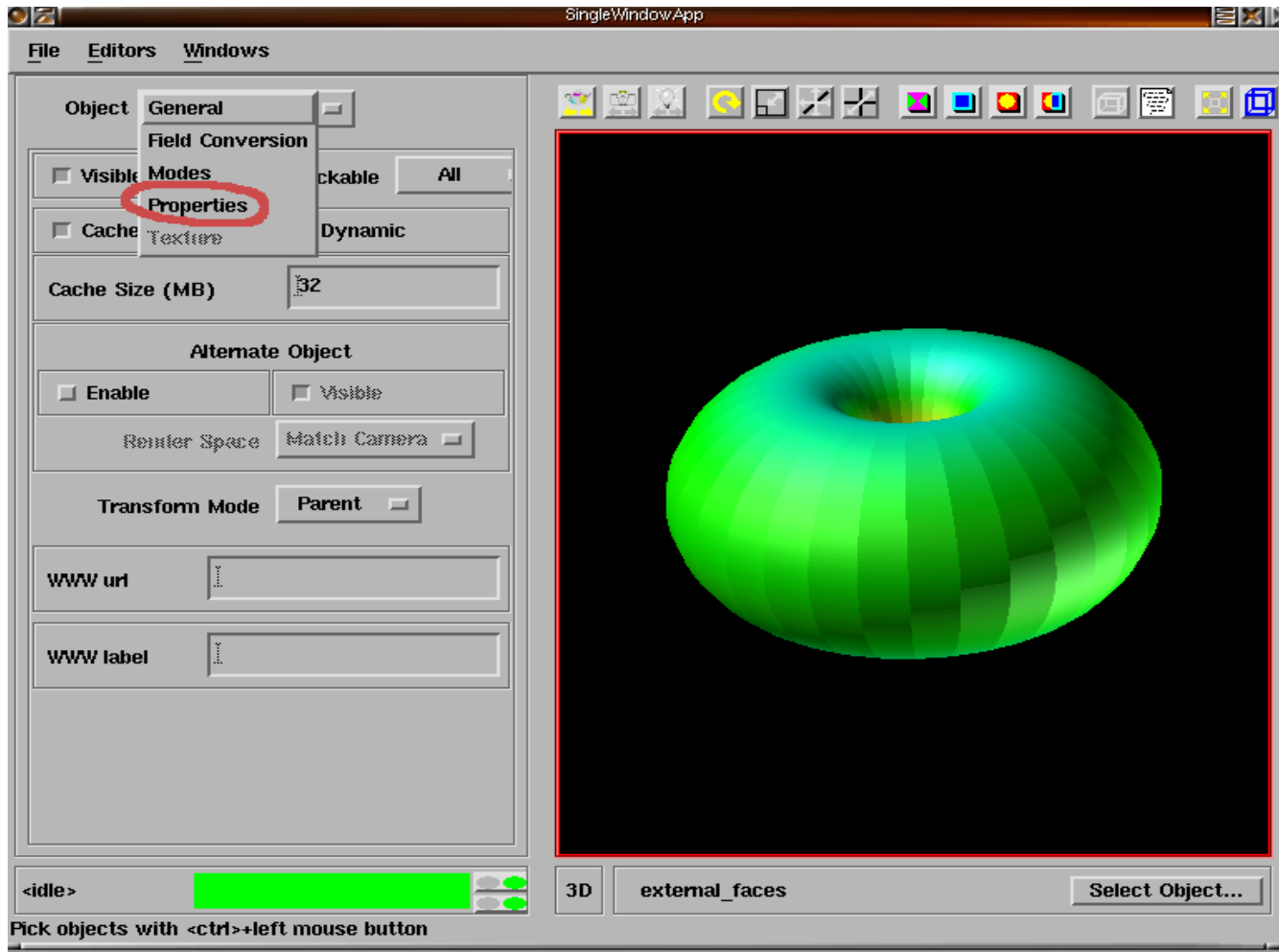
ctrl-left click on the
surface to select
the object.

external_faces
will replace Top.

Select
Editors/Object
from the main
menu.

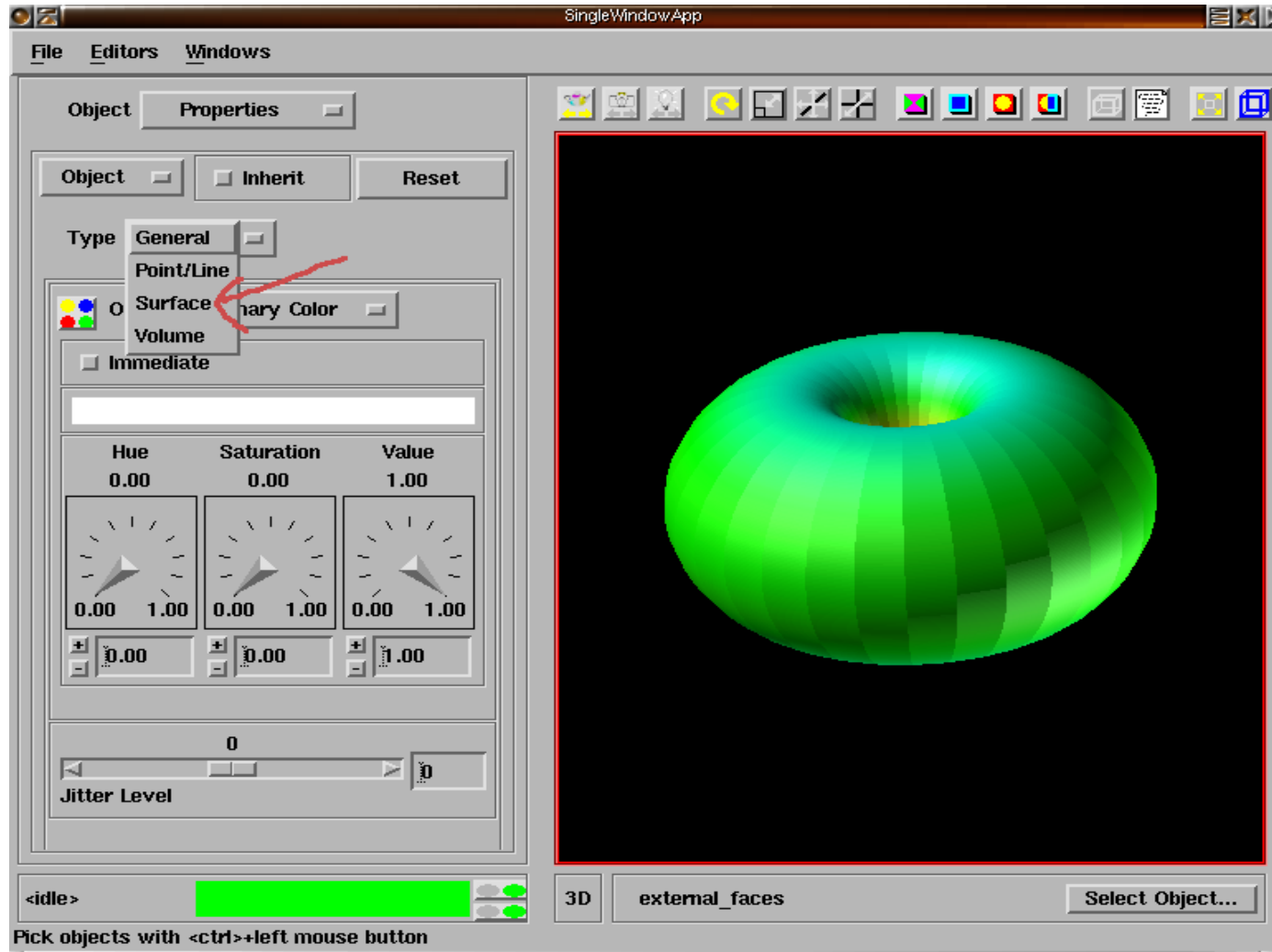


Example Application II



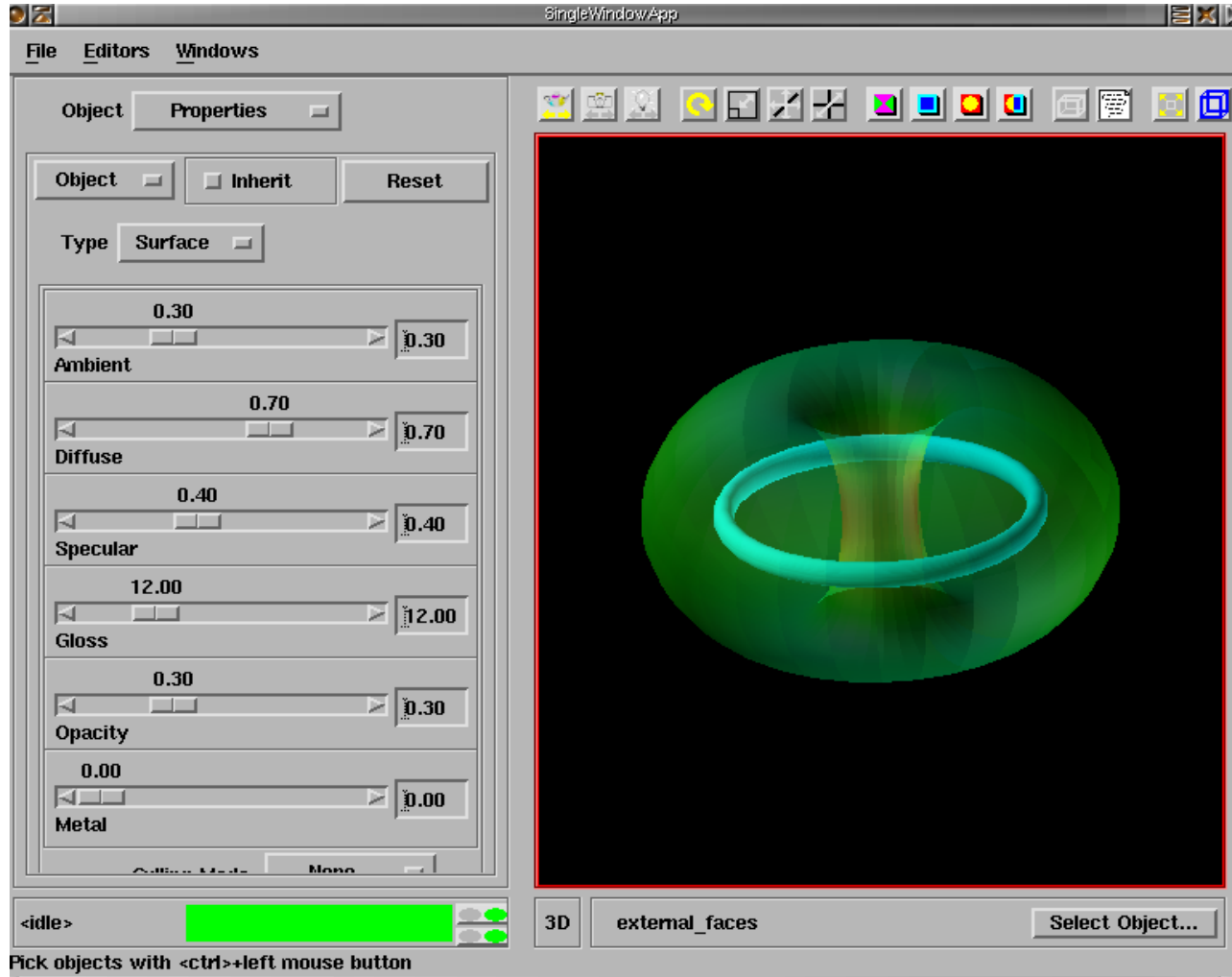


Example Application II



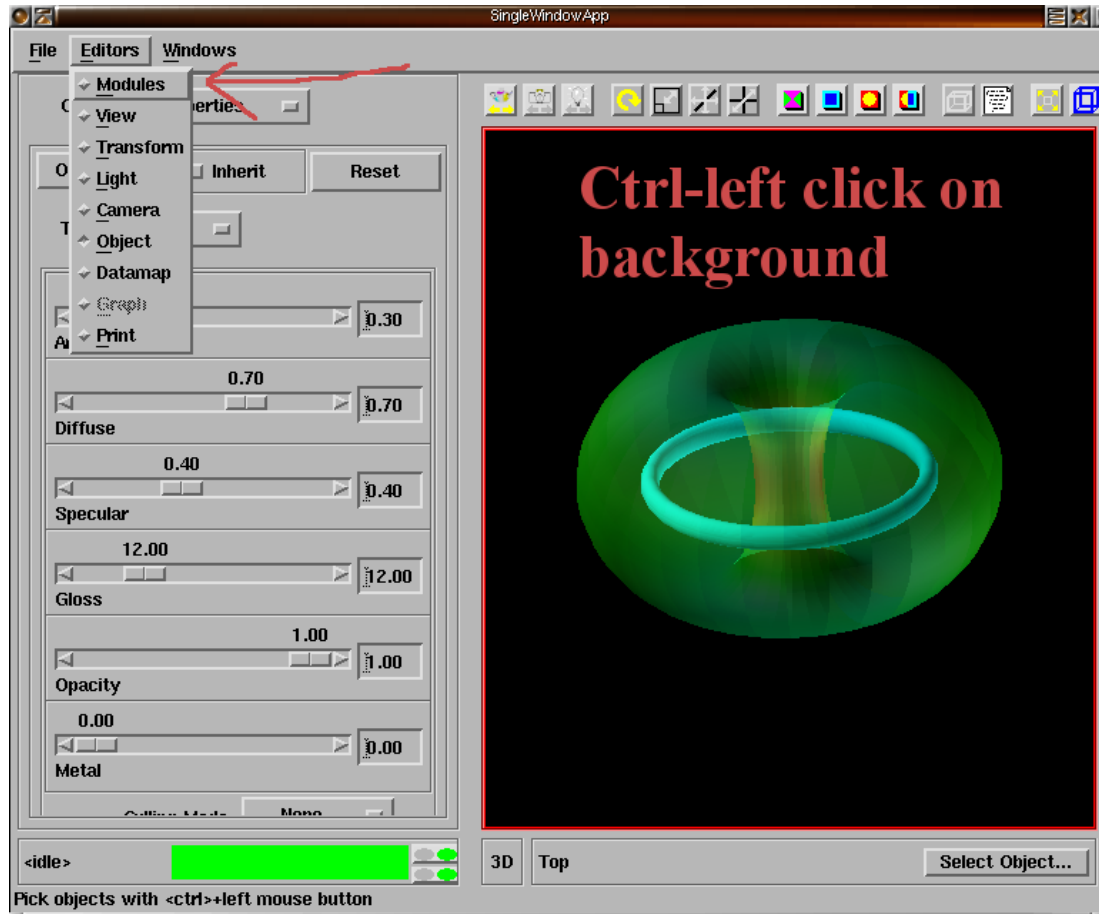


Example Application II





Example Application II

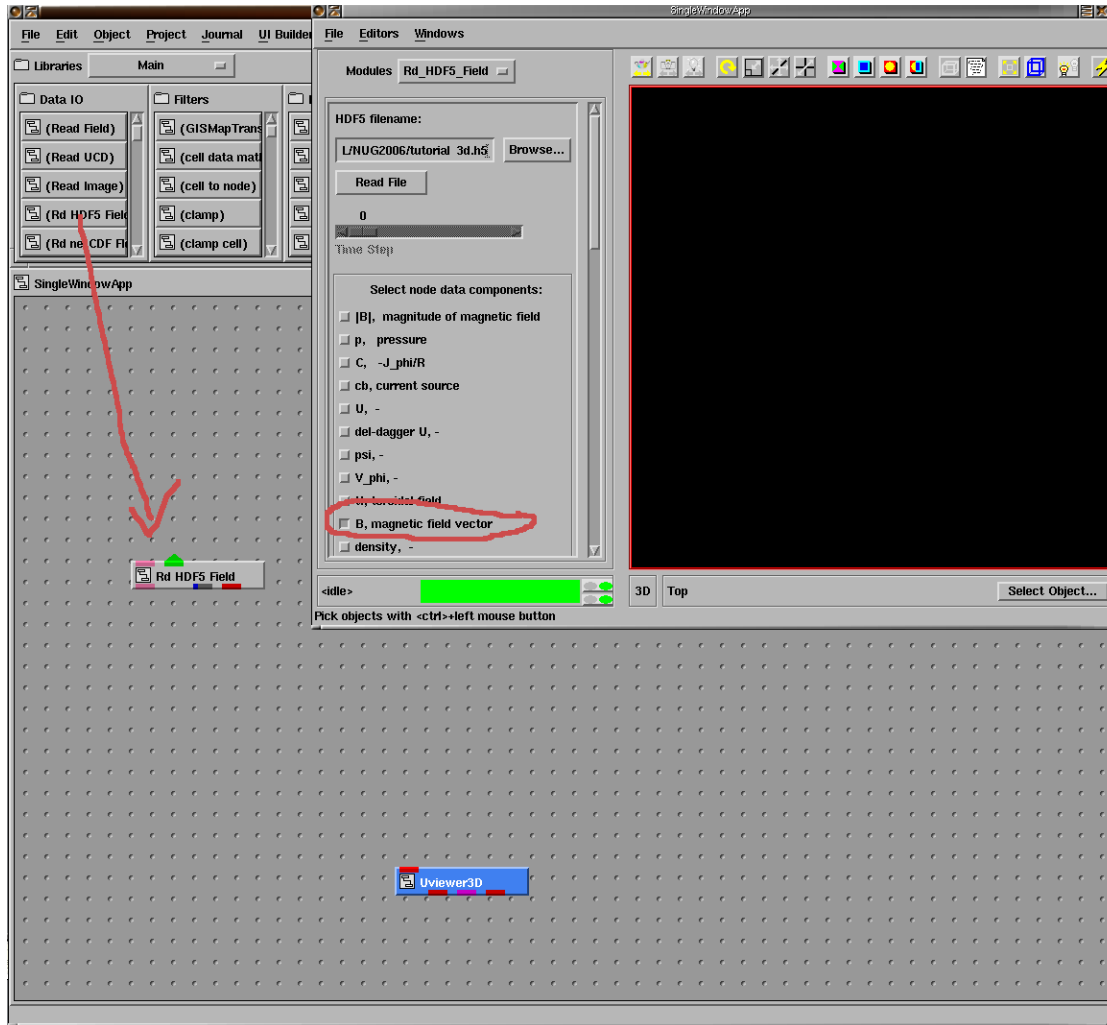


ctrl-left click on the background to select the Top object.

Select Editors/Modules from the main menu.



Example Application III

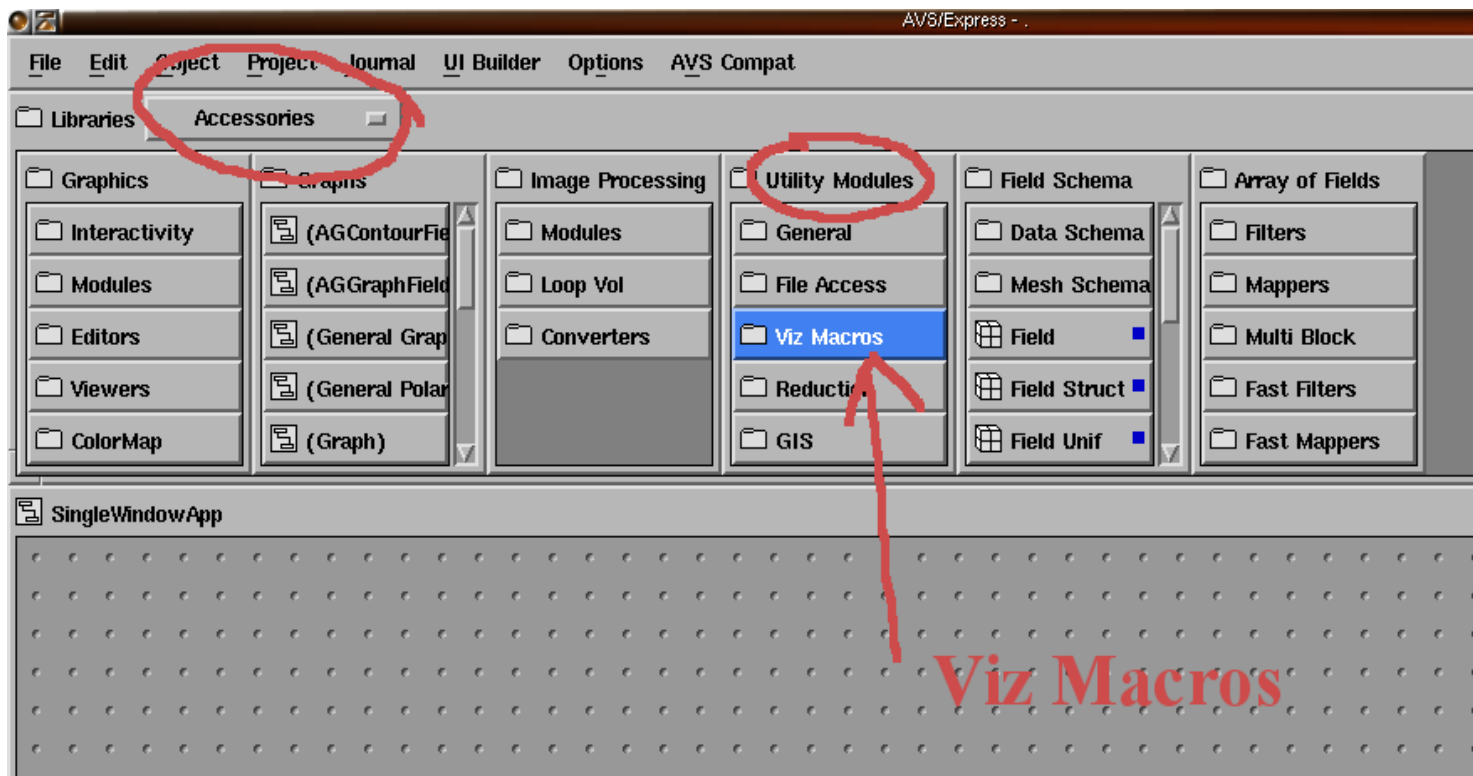


We will visualize a Vector Field:

From the Rd_HDF5_Field GUI read the

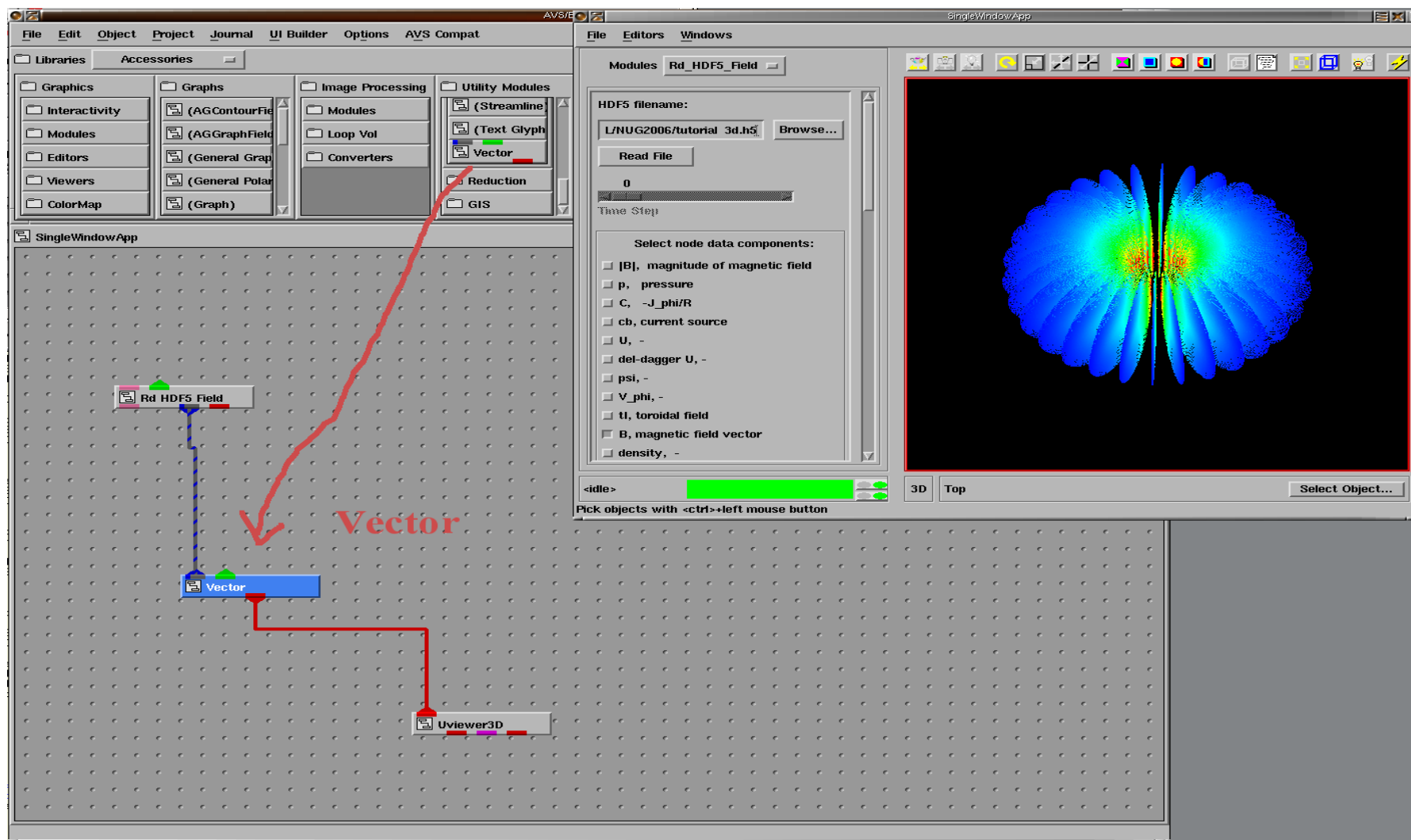
B, magnetic field vector

Example Application III



**Select
Accessories/Utilities/Viz_Macros/Vector
(double left click to open)**

Example Application III



The screenshot displays the AVS/Express software interface. On the left, a workflow diagram shows the connection between three objects: 'Rd HDF5 Field' (top), 'Vector' (middle), and 'Uviewer3D' (bottom). A red arrow points from the 'Vector' object to the 'Rd HDF5 Field' object, and another red arrow points from the 'Uviewer3D' object to the 'Vector' object. A large red arrow labeled 'Vector' points from the 'Vector' object towards the 'Rd HDF5 Field' object.

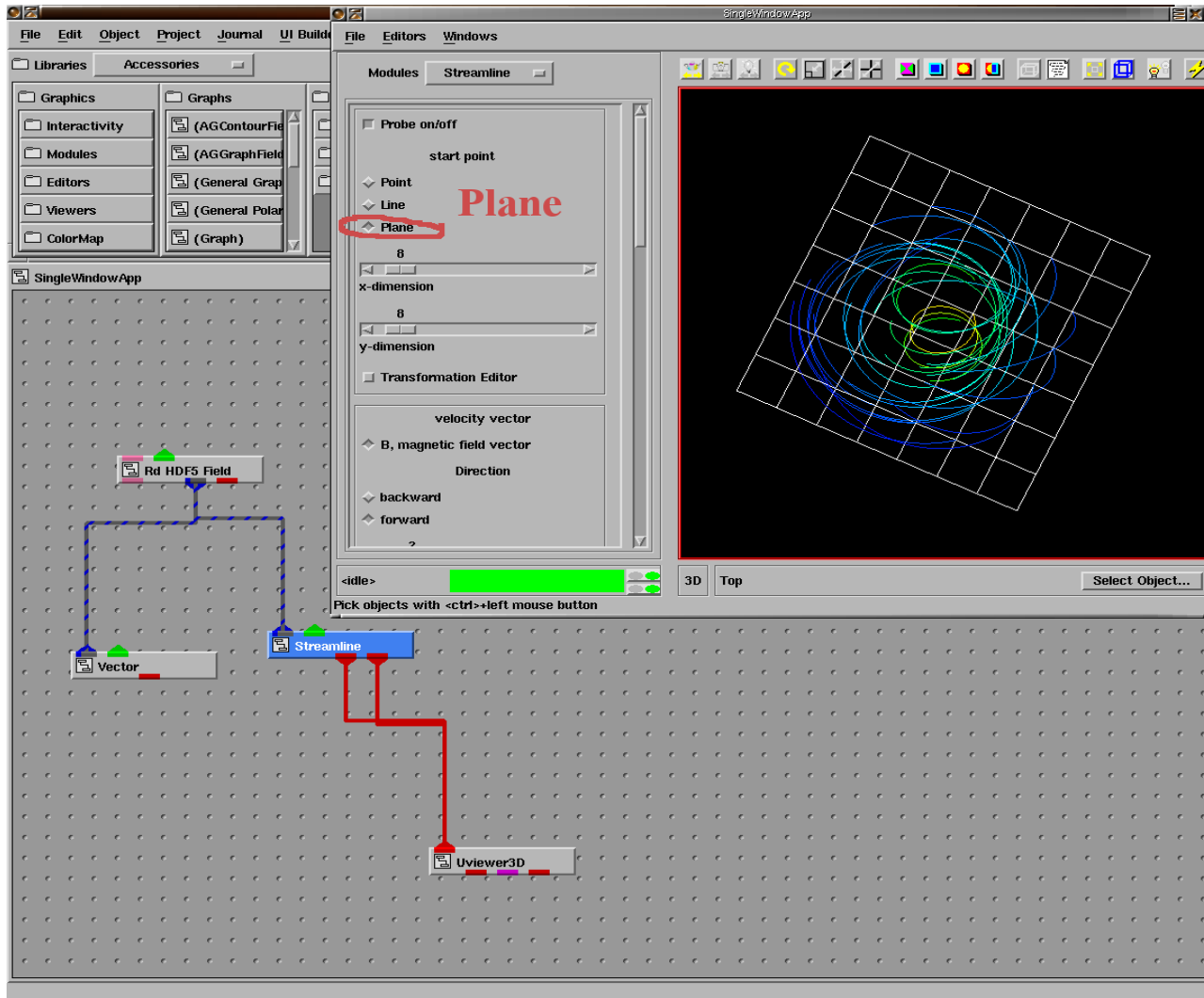
The central panel shows the 'Modules' for 'Rd_HDF5_Field'. The 'HDF5 filename' is set to 'L/NUG2006/tutorial_3d.h5'. The 'Time Step' is set to '0'. Under 'Select node data components', the following options are listed:

- |B|, magnitude of magnetic field
- p, pressure
- C, -J_phi/R
- cb, current source
- U, -
- del-dagger U, -
- psi, -
- V_phi, -
- ti, toroidal field
- B, magnetic field vector
- density, -

The right panel shows a 3D visualization of the magnetic field vector field, rendered as a blue and green toroidal structure. The bottom right corner of the interface includes a '3D Top' view selector and a 'Select Object...' button.



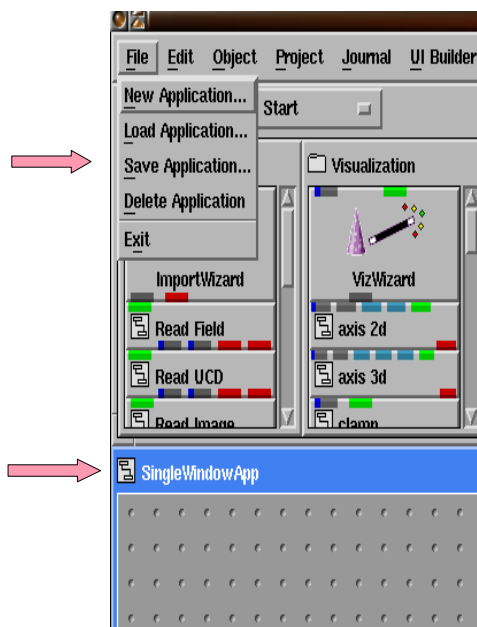
Example Application III



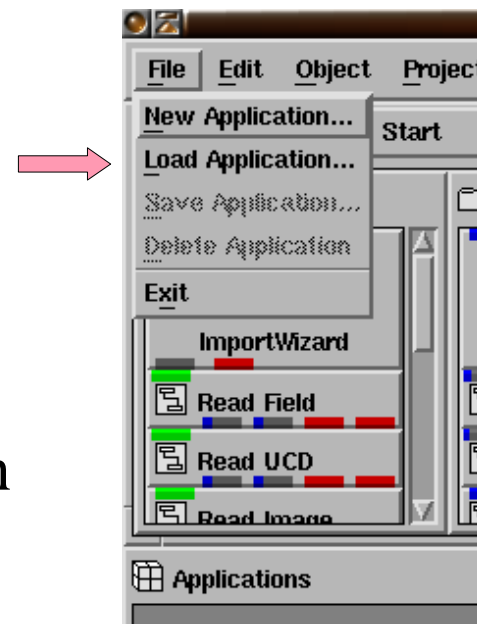
Delete the vectors from the viewer and connect the streamlines.

Select plane as the start point

AVS/Express applications can be saved as .v files and loaded later.



To save an application:
select your
SingleWindowApp (left
click)
and save from the File
menu as a .v file



You can load it later with
“File/Load Application”



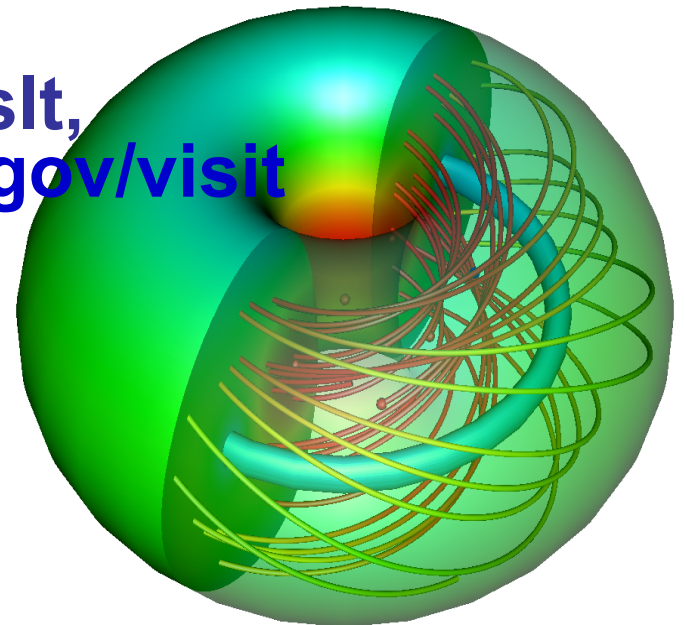
AVS/Express Module Writing

It's a long tutorial by itself. If people are interested I can prepare material and post it in our web site.

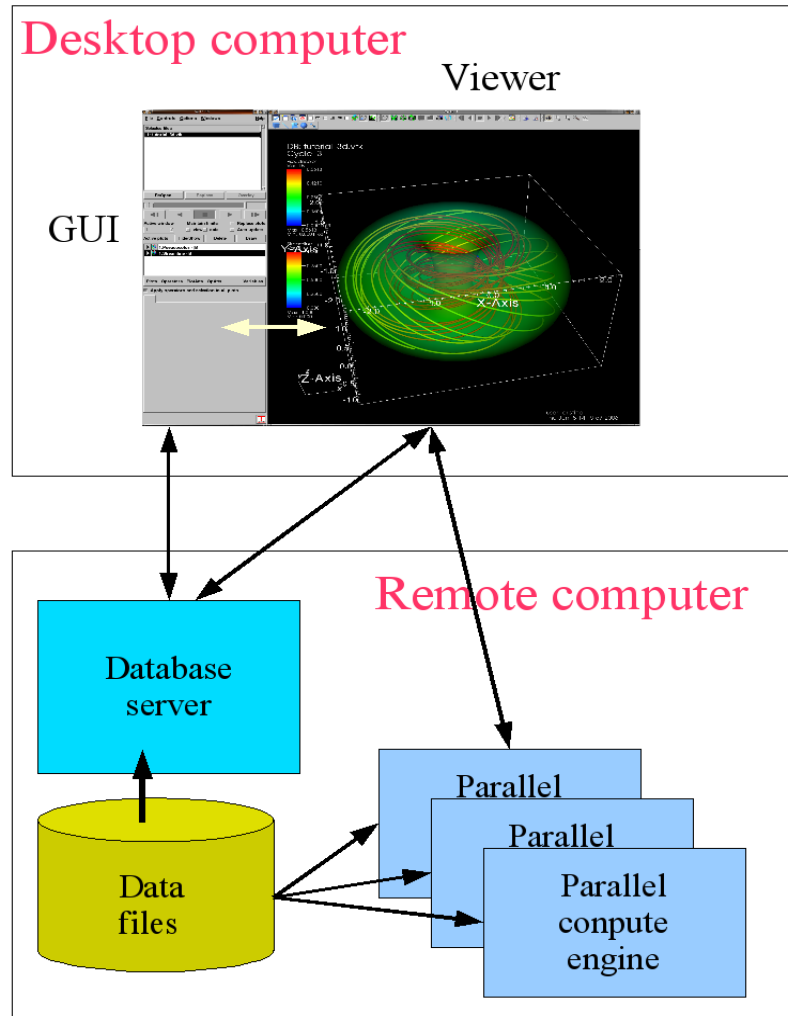


Visit tutorial

- **Visit is a point-and-click 3D scientific visualization application that supports most of the common visualization techniques (isocontouring, volume rendering) on structured and unstructured grids.**
- **For detailed information on Visit, please refer to <http://www.llnl.gov/visit>**



- **Main Features:**
 - it's free and open source and most importantly, it offers a **distributed** mode. You can connect a client from your desktop and run a backend in the computer where you produced the data.
 - VisIt employs a **parallel** architecture in order to handle extremely large data sets interactively.





Visit tutorial

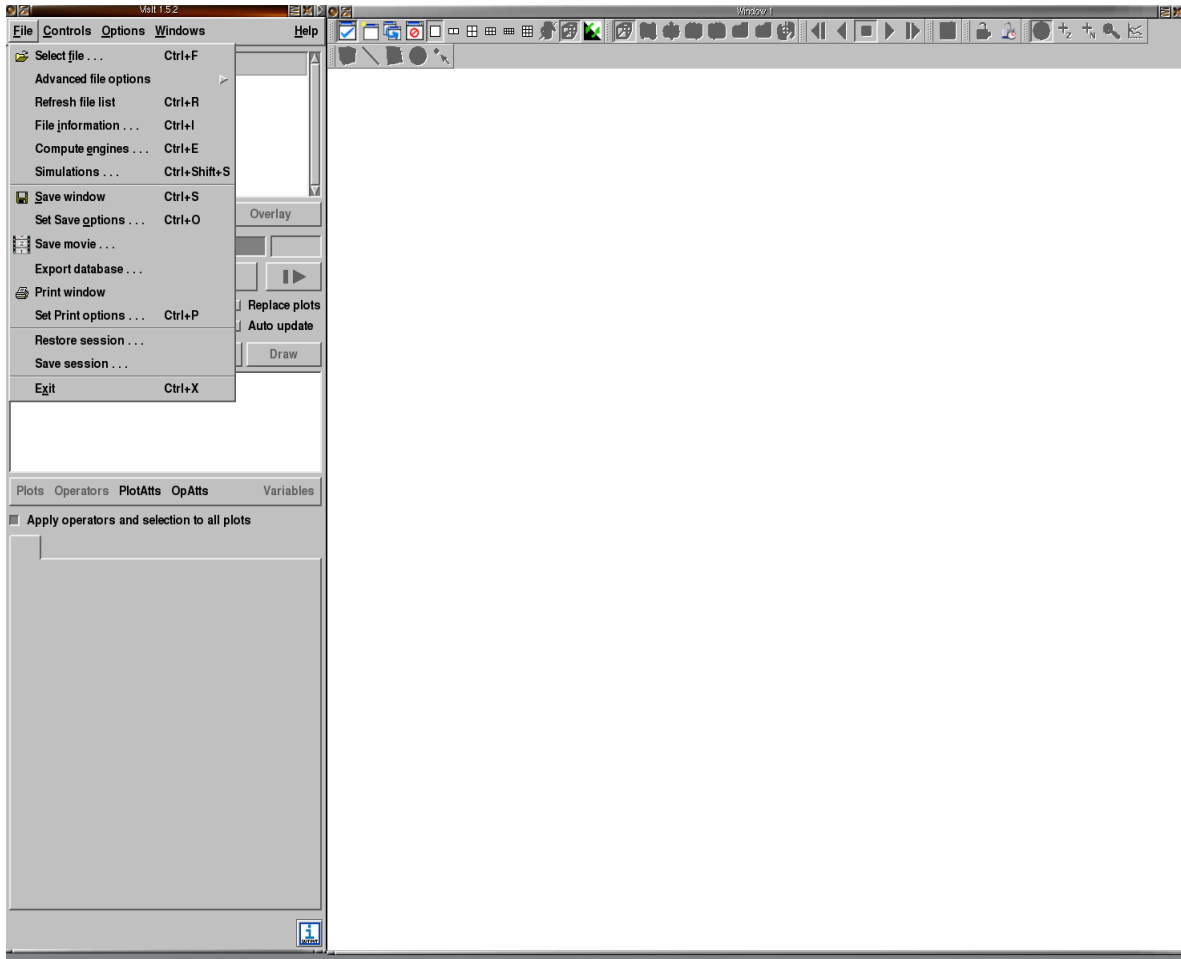
- On DaVinci, use the modules facility
% module load visit
% visit

IMPORTANT NOTES

- 1- To run the parallel backend the path to visit has to be in your environment. Add this line to your .cshrc
setenv PATH /usr/common/graphics/visit/visit-1.5.2/bin/:\$PATH
- 2 - Sometimes when the Visit GUI starts, the opening of windows stalls. If this is the case use:
 - %visit -nowindowmetrics
- 3 - If you are running in client - remote server mode, the version number of the client and the server must to be the same.
- 4- Make sure the backend can connect to your client (check with the systems' admin).



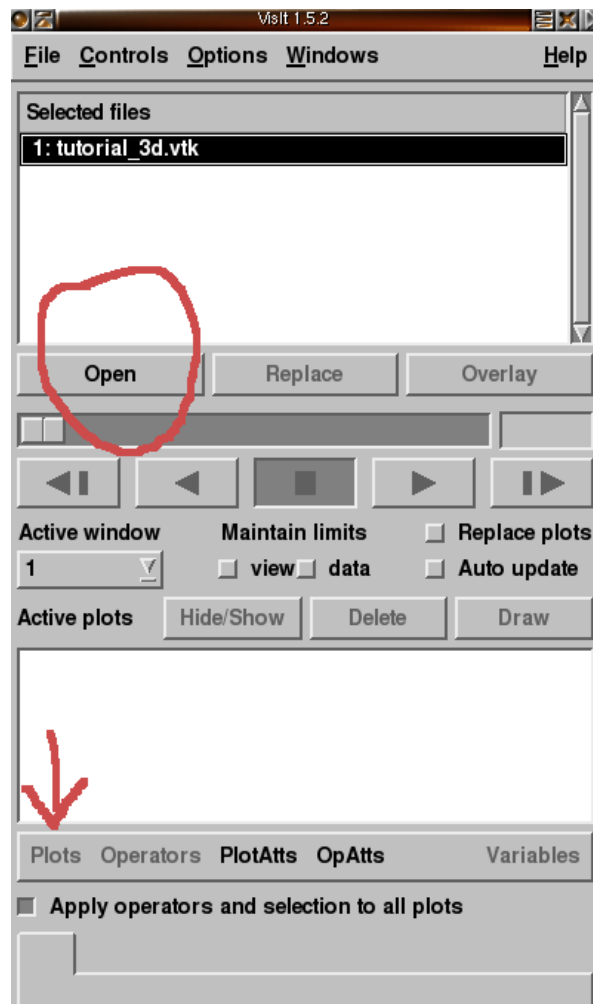
Visit tutorial



Select
“Select file...”
from the File menu.

Load
“tutorial_3d.vtk”

Note: this is the
same data that
we just used for
Express
but I extracted only
the B, magnitude
and the
B, vector variables
and converted them
to VTK format.

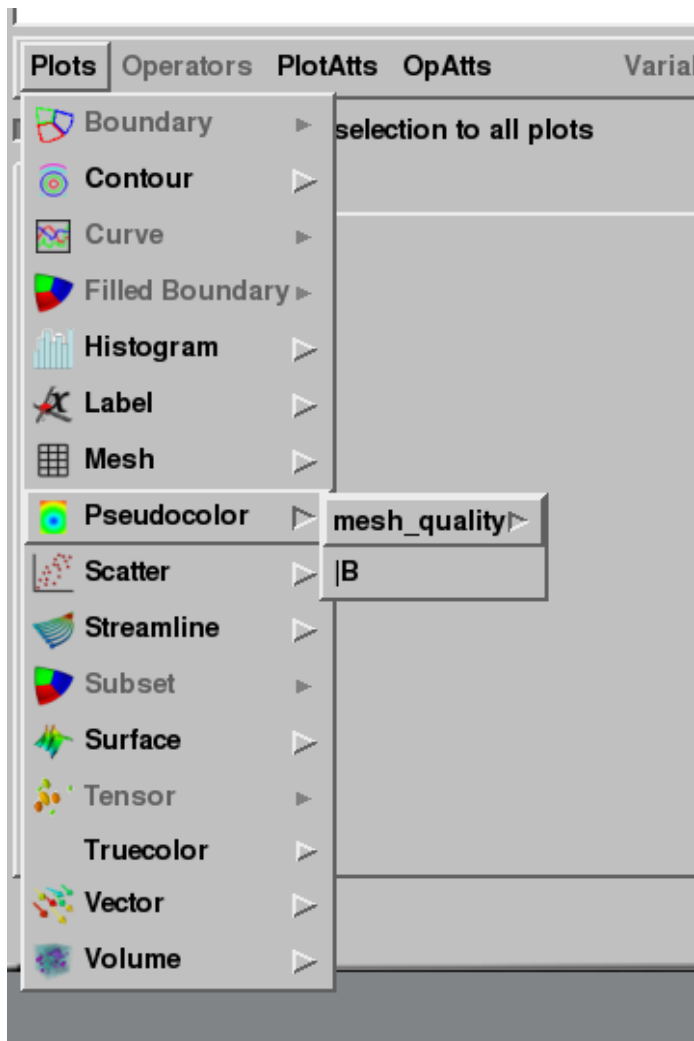


Open the file.

The Plots menu will be enabled.

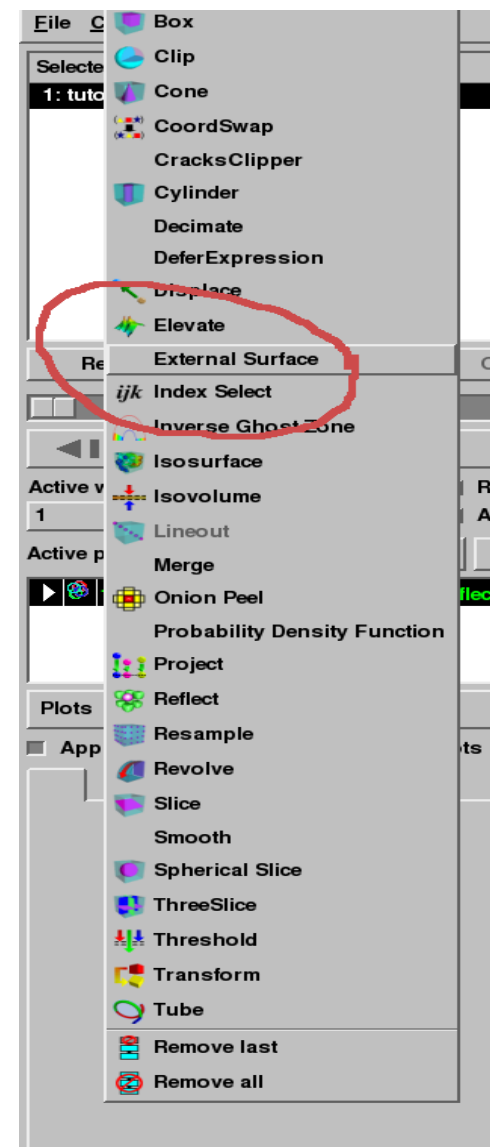
Once you select a plot, the operators over that plot will be enabled.

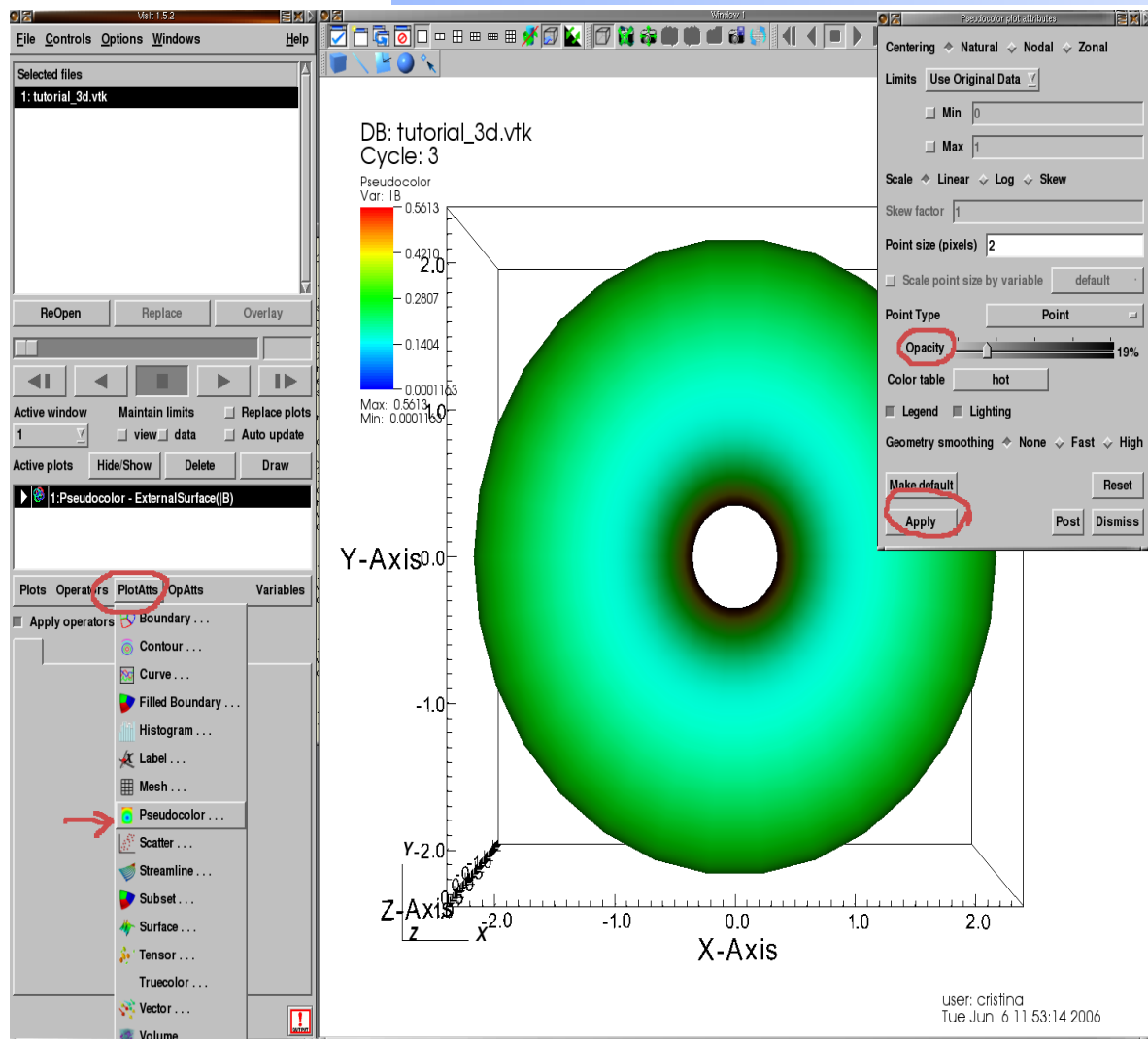
Visit tutorial



Select Plots,
Pseudocolor, |B

Operators will be
enabled, Select
External Surface

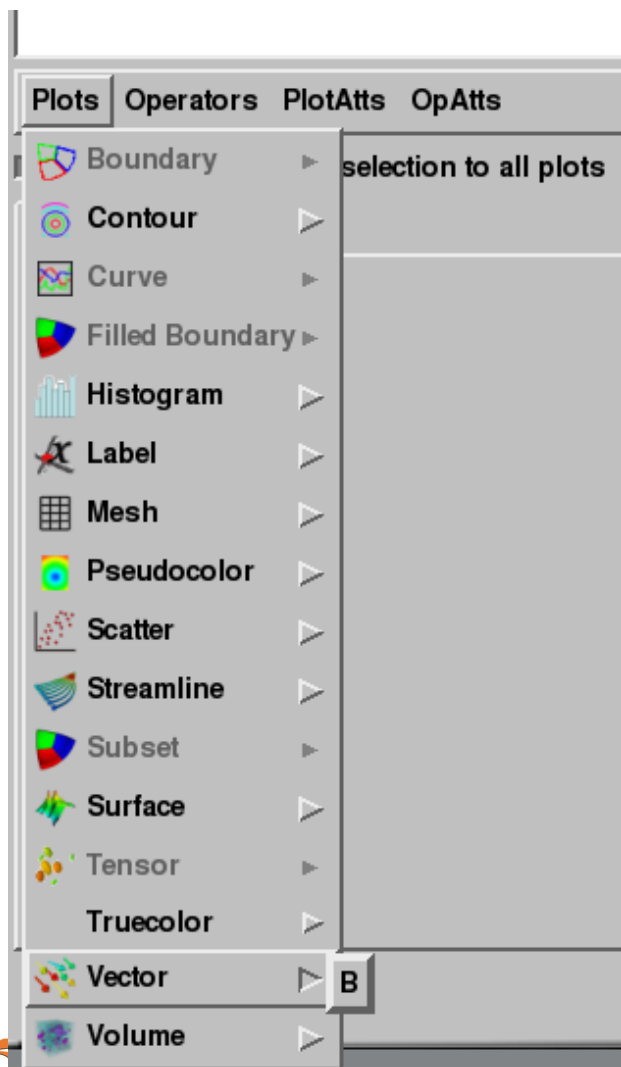




Select PlotAtts
Pseudocolor.

Change the
opacity and
apply.

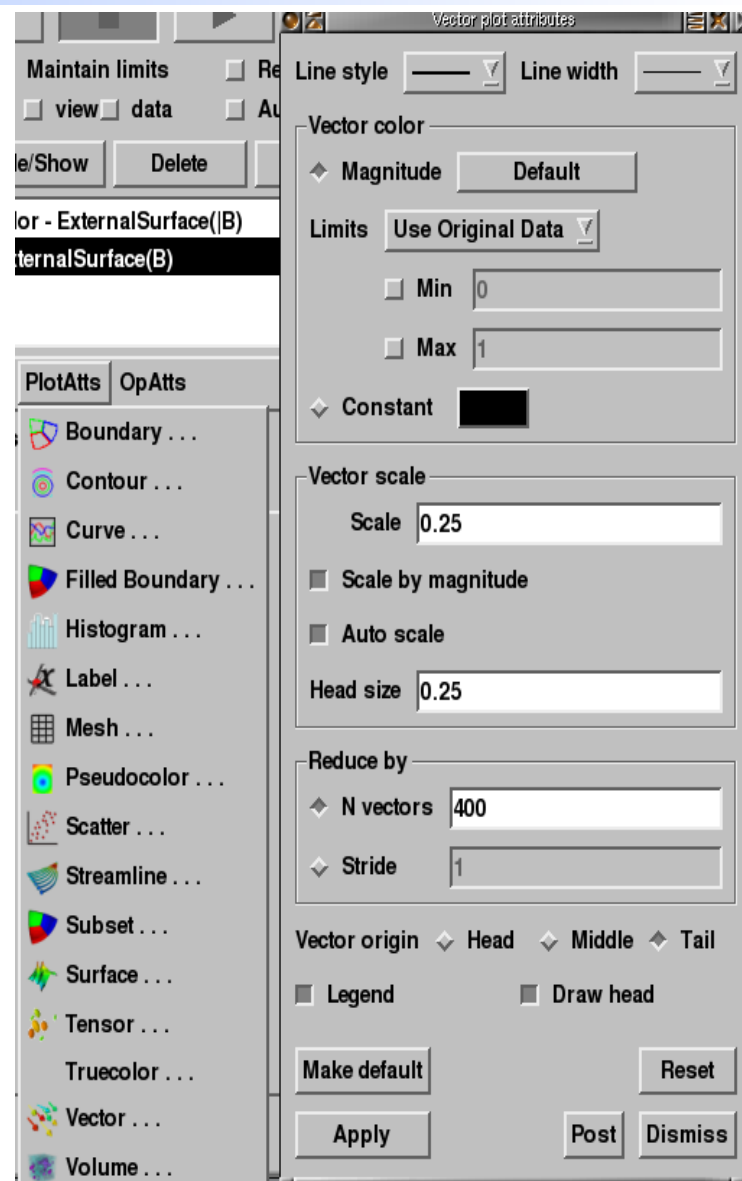
Visit tutorial



Select Plot
Vector B

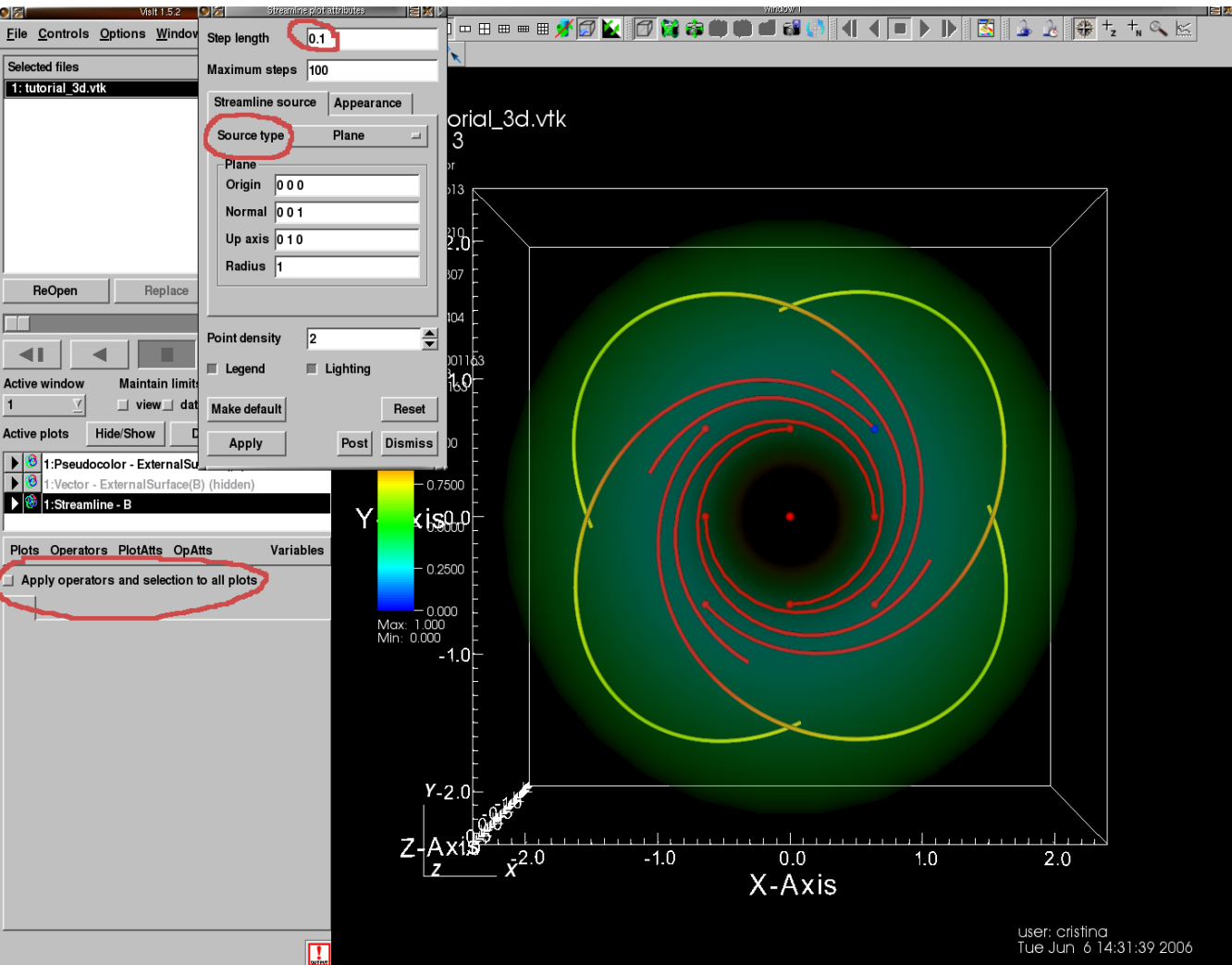
Draw the plot

From the
PlotAtts menu
select Vector
and change the
parameters





VisIt tutorial



Making Streamlines:

Make sure you deselect the toggle button “Apply operators and selection to all plots”

Select Plots, Streamline, B

From PlotAtts, Streamline change the source type to plane and the step length.



Conclusions

- **AVS/Express: commercial, serial, very flexible, not so easy to use, good API for custom development.**
 - left out: module development
- **VisIt: open source, parallel, easy to use, python interface for batch operation, plugin development is possible.**
 - left out: vtk data format, parallel operation, plugin development.