



# **pyCGNS.NAV/Manual**

***Release 4.0.1***

**Marc Poinot**

May 13, 2011



# CONTENTS

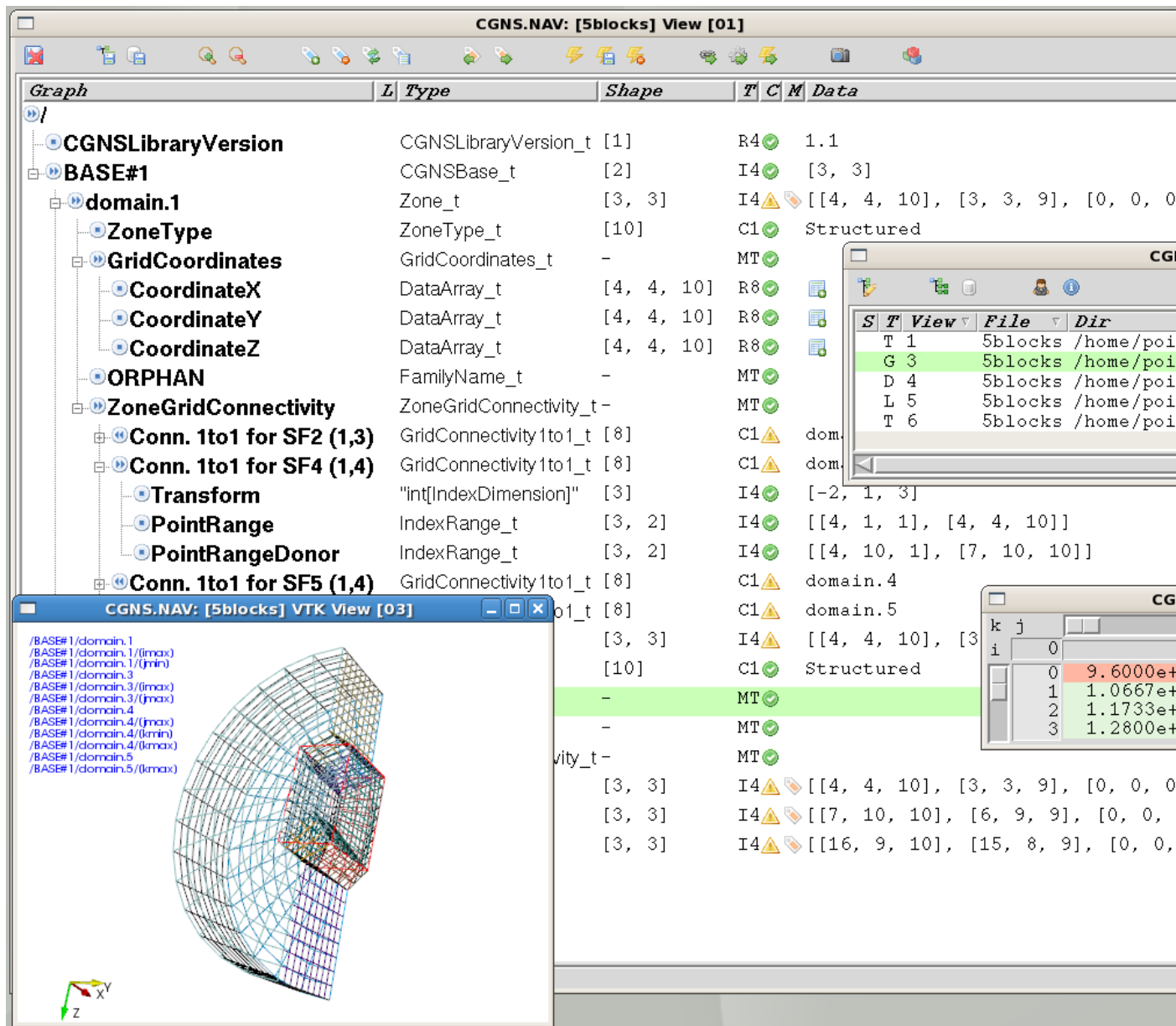
<b>1</b>	<b>QuickStart</b>	<b>3</b>
<b>2</b>	<b>Tree View</b>	<b>7</b>
2.1	The icon menu . . . . .	9
2.2	Selection and flags . . . . .	9
2.3	Copy/Paste . . . . .	9
<b>3</b>	<b>Option View</b>	<b>11</b>
3.1	Options . . . . .	11
3.2	Default key/mouse bindings per view . . . . .	12
<b>4</b>	<b>Pattern View</b>	<b>15</b>
<b>5</b>	<b>VTK View</b>	<b>17</b>
5.1	Camera . . . . .	17
5.2	Key bindings . . . . .	17
<b>6</b>	<b>Query View</b>	<b>19</b>
<b>7</b>	<b>Link View</b>	<b>21</b>
<b>8</b>	<b>Table View</b>	<b>23</b>
<b>9</b>	<b>NAV Index</b>	<b>25</b>



If you want to browse your CGNS file, just type:

```
CGNS . NAV
```

and the pyCGNS browser appears.



The file format is detected using the file extension. You can give a list of files to CGNS.NAV, it opens each file w.r.t. its type. The .cgns extension uses the CGNS library which can detect both CGNS/ADF and CGNS/HDF formats:

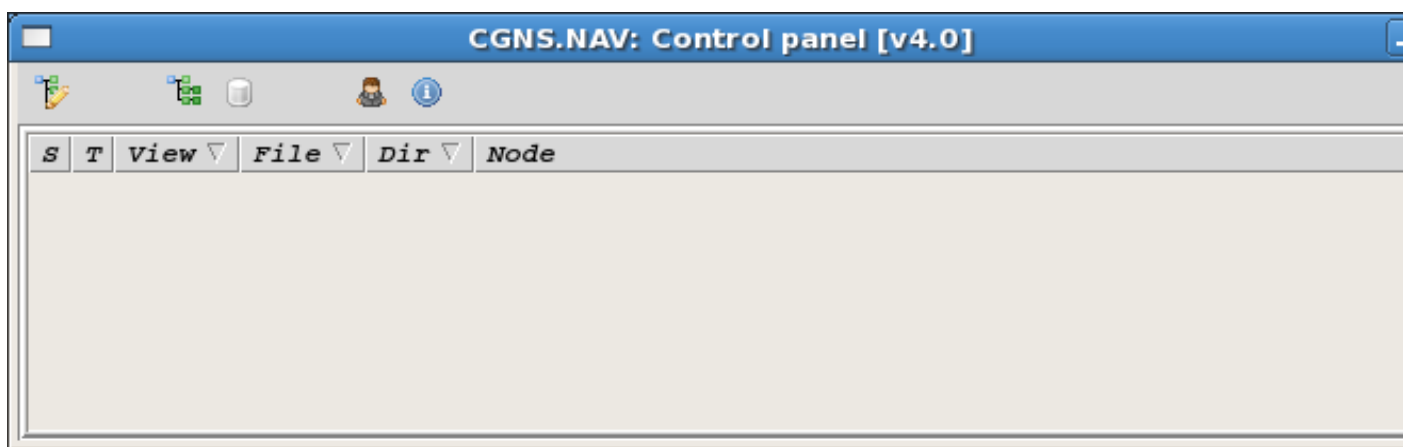
```
CGNS.NAV wing.cgns plane.adf helicopter.hdf cror.py
```

It can load/save CGNS files with the HDF5/ADF and Python formats, parse and display the contents, edit the contents using simple edit and copy/paste commands, select nodes using complex queries, use already defined patterns such as the *SIDS* patterns and other interoperability features.



# QUICKSTART

The first window you see when you start `CGNS.NAV` is the *Control View*. This main view helps you to find out the views you have and on which CGNS tree they refer to.

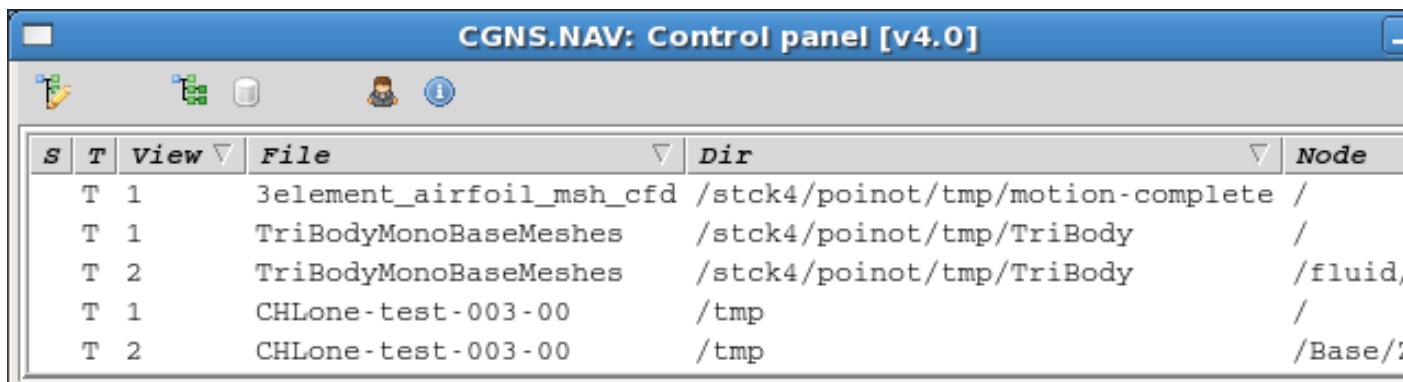


If you select the line of an existing view in the *Control view* and press *Return*, this view will be raised on your screen. You can find back the *Control view* from any other view just pressing the *BackSpace* key (See [Key/Mouse Bindings](#)).



From its icon menu you can *create* a new CGNS/Python tree, you can *Load* an existing file or open the *Pattern View*. There is also an icon for the *Option View* and an info about `CGNS.NAV` version.

Each time you open a view on a file, the *Control* keeps track of it, you can check all the views you may have on the same file.



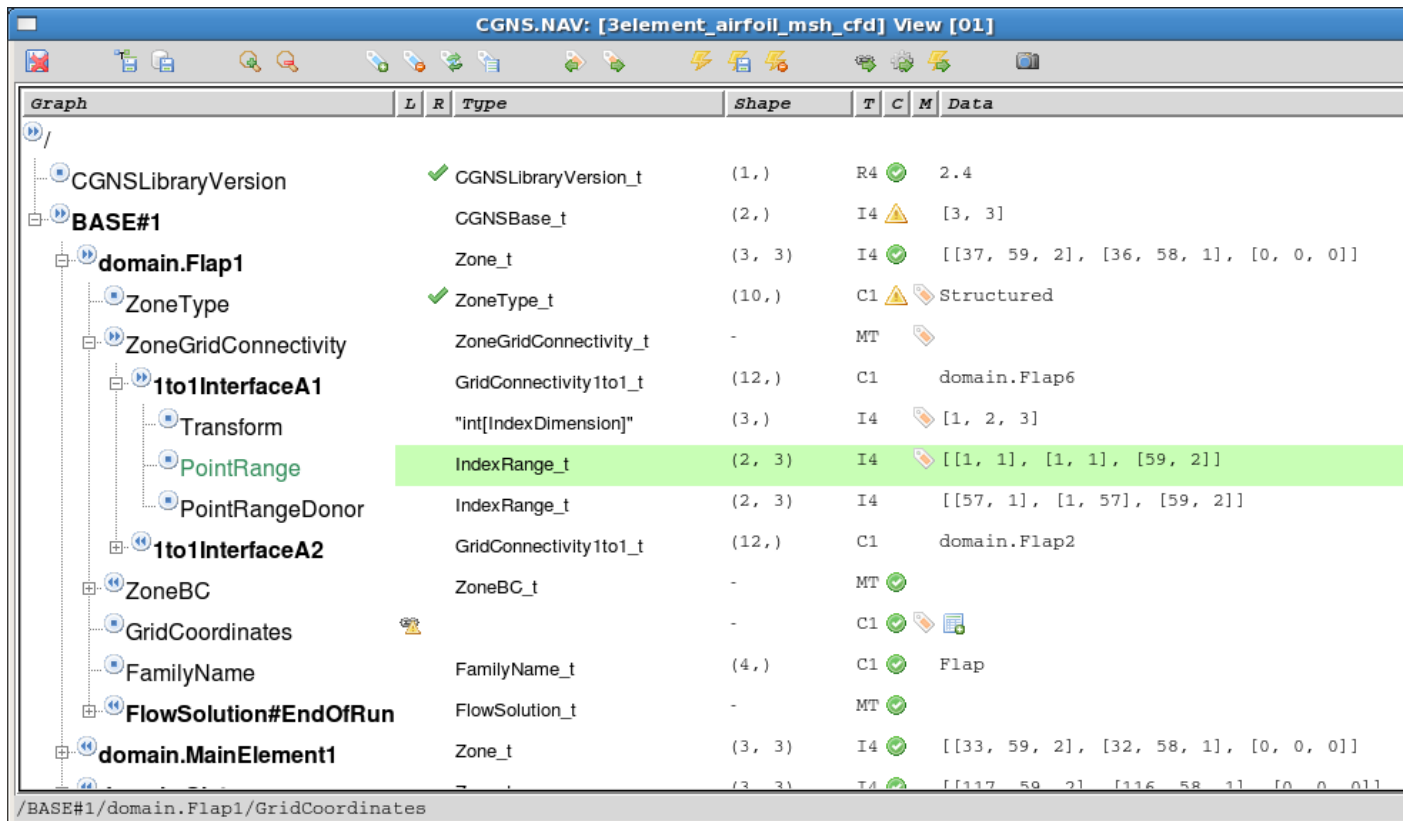
S	T	View	File	Dir	Node
T	1		3element_airfoil_msh_cfd	/stck4/poinot/tmp/motion-complete	/
T	1		TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/
T	2		TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/fluid,
T	1		CHLone-test-003-00	/tmp	/
T	2		CHLone-test-003-00	/tmp	/Base/

The columns of the lines you see in this *Control* view are:

- *S* status flag, says if you tree has been modified and needs a save
- *T* type of view, in this case you only have *tree* views

- The number of the view if you have many views for the same CGNS tree
- The file name (when you create from scratch we propose a default name)
- The directory where you loaded/saved the file
- The root node of the view

The *Tree View* appears when you *create* a new tree or when you *open* an existing one. The tree below is a large one and we have a lot of information.



You click on the +/- icon before any node name in the tree to open/close the child sub-tree for this node. A mouse left button simple click on a node selects the node. A mouse right button simple click opens the menu for the selected node. The menu gives you access to some function such as Copy/Paste, change node name, type, add a child... Most of these functions have keyboard shortcuts.



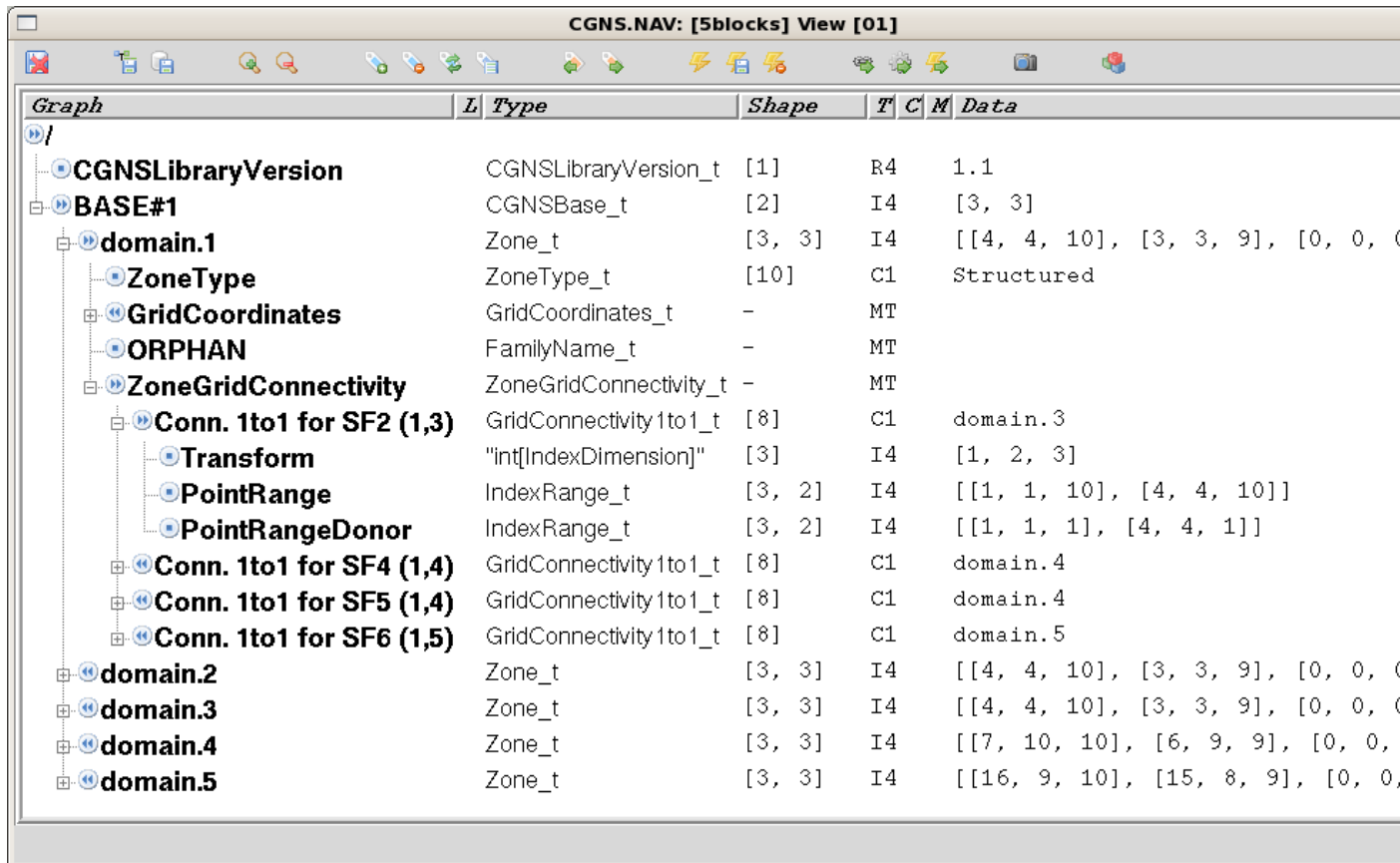
...BASE#1	CGNSBase_t	(2,)	I4	[3, 3]
domain.Flap1	domain.Flap1	(3, 3)	I4	[[37, 59]
ZoneType	Change name (C-a)	(10,)	C1	Structure
ZoneGridConnectivity	Change CGNS type	-	MT	
1to1InterfaceA1	Change CGNS type (C-s)	(12,)	C1	domain.F
Transform	Change data type	(3,)	I4	[1, 2, 3]
PointRange	Change value (C-e)	(2, 3)	I4	[[1, 1],
PointRangeDonor		(2, 3)	I4	[[57, 1]
1to1InterfaceA2	Add link (C-l)	(12,)	C1	domain.F
ZoneBC	Remove link	-	MT	
GridCoordinates	Copy (C-c)	-	C1	
FamilyName	Cut (C-x)	(4,)	C1	Flap
FlowSolution#EndOfRun	Paste as brother (C-v)	-	MT	
	Paste as child (C-y)	(3, 3)	I4	[[33, 59]
domain.MainElement1		(3, 3)	I4	[[117, 59]
domain.Slat	Open tree view (C-w)	(3, 3)	I4	[[29, 59]
domain.Flap2	Open table view (C-t)	(3, 3)	I4	[[15, 59]
domain.Flap3		(3, 3)	I4	[[13, 59]
domain.Flap4	Add brother (C-b)	(3, 3)	I4	[[25, 59]
domain.Flap5	Add child (C-u)	(3, 3)	I4	[[57, 59]
domain.Flap6		(3, 3)	I4	[[55, 53]
domain.MainElement2	Check (C-z)	(3, 3)	I4	
	Zone_t			



# TREE VIEW

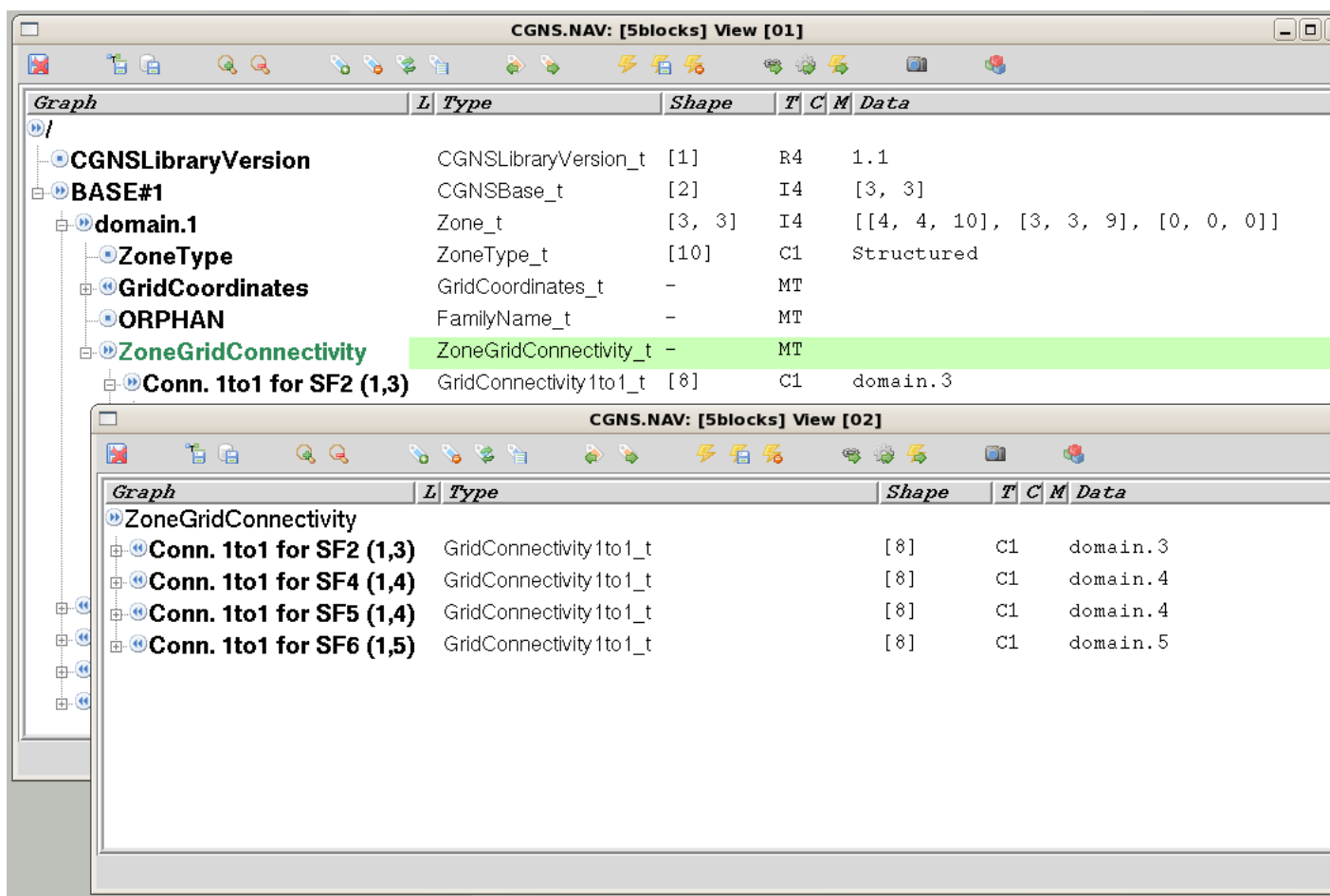
The first window `CGNS.NAV` opens when you read a *CGNS* file is the *Tree view*. This is your main view of the tree, the browsing or modification of your tree has to be performed in this view or in a sub-window created from this view.

The *Tree view* shows you the tree of nodes in a table-like way, you can see at the same time many informations on a complete sub-tree.

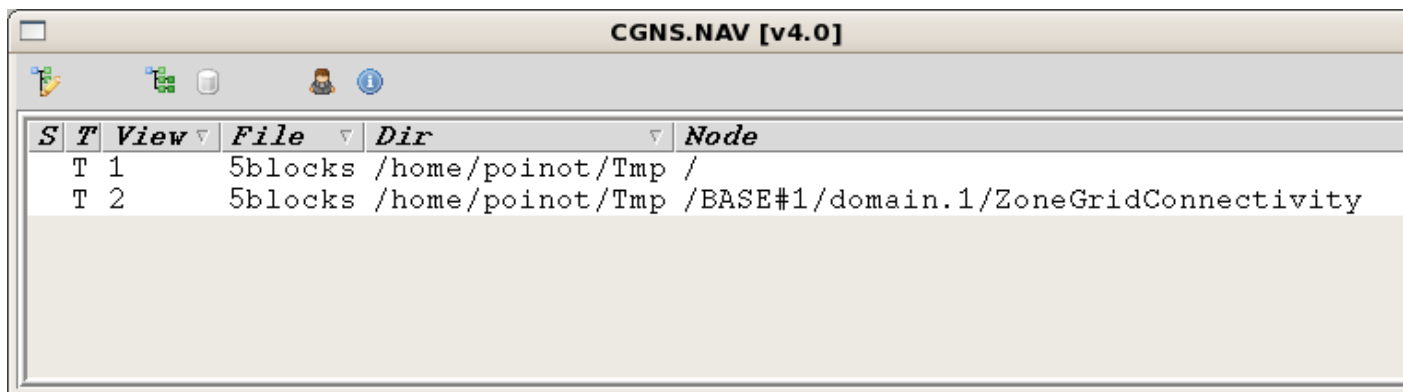


A *Tree view* can be a view of a sub-tree of your current view. Select the top node you want as sub-tree root, press `<Control-W>` and the new window appears (see <sup>1</sup>).

<sup>1</sup> All key bindings can be customized, the bindings we give here are the *default bindings*.



You can track with the *Control* view which window you have on which node. To find back the *Control* view when you are in a *Tree* view, press <BackSpace>.



**Note:** The sub-tree window is another view of the **same** CGNS tree. If you change something in one window, all existing windows displaying the same tree are updated.

## 2.1 The icon menu

## 2.2 Selection and flags

## 2.3 Copy/Paste

The *Tree* view support the *Cut/Copy/Paste* on trees. You can copy a sub-tree in a view, paste it into another view (even a view that belongs to another *CGNS* tree). The usual key bindings are `<Control-X>` for *Cut*, `<Control-C>` for copy and `<Control-V>` for paste.

When you *Cut/Copy*, the selected node uses as the root node of your buffered tree. When you *Paste*, the selected node can act as the parent node of the new sub-tree (using `<Control-Y>`) or as the brother of your new sub-tree (using `<Control-V>` (see <sup>2</sup>).

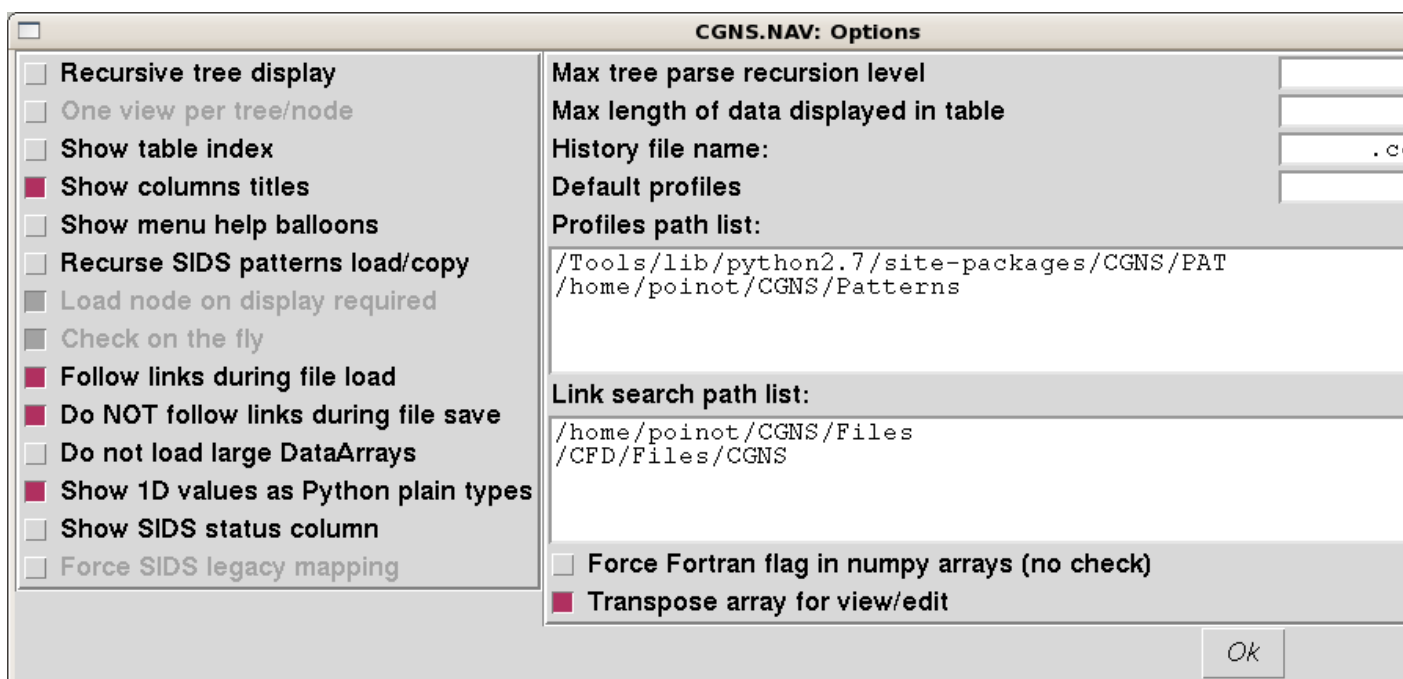
---

<sup>2</sup> A *Cut* or a *Copy* action puts the selected root node path into a buffer. There is a single buffer for all *CGNS.NAV*, so if you *Cut* or *Copy* several times only the last one is taken into account. A *Copy* actually copies the sub-tree starting from the path in the buffer, recursively up to the end of all branches. The root node of the resulting copy is



## OPTION VIEW

The *Option view* is the place where you set your own parameters for *CGNS.NAV*. You should be sure the right options are set because some of these are very important and could change the actual contents of a file you want to store.



You can change values in the *Option view* or edit the `.cgnsnavoptions.py` file you can find in your home directory. For example you can customize the key bindings only in the `.cgnsnavoptions.py` file. This file is a plain Python file, it is imported and re-generated each time you run `CGNS.NAV`.

The `.cgnsnavhistory.py` contains the history of directories where you opened a *CGNS* file.

### 3.1 Options

The check-box options are active when the box is *checked* (wouahahah !). Some are only related to the display of the data, some other are driving the way *CGNS.NAV* reads or writes the data on disk and have a strong impact on the actual *CGNS* file you will obtain.

- ***Recursive tree display forces the Tree view to open all the children*** recursively. We strongly suggest you *un-check* this box on large files.
- *One view per tree/node* (Not available) when active you cannot open more than one view per tree/node.
- *Show columns titles* is used to show/hide the top titles of tables

- *Show menu help balloons* activates ugly yellow balloons giving a hint on the purpose of the icons in the views menus.
- *Recurse SIDS patterns load/copy* indicates that you want to load or copy the SIDS CGNS sub-trees in a recursive way. For example, if you load the *CGNSBase\_t* pattern you will obtain the whole SIDS pattern, all pattern children of *CGNSBase\_t* will be loaded as well.
- *Load node on display required* (Not available) to save memory and time, CGNS node is read from the file only when *CGNS.NAV* needs to display it.
- *Check on the fly* (Not available) forces a check each time you modify a tree.
- *Follow links during file load* indicates that the links *CGNS.NAV* encounters into a CGNS file have to be followed or not.
- *Do NOT follow links during file save* means that the links will be ignored during save which has the effect to *merge* all the tree nodes into the same root file.
- *Do not load large DataArrays*
- *Show 1D values as plain Python types*
- *Show SIDS status column*
- *Max tree parse recursion level*
- *Max length of data displayed in table*
- *History file name*
- *Default profile*
- *Profile path list*
- *Force fortran flag in numpy arrays*
- *Transpose array for edit/view*

## 3.2 Default key/mouse bindings per view

The *<C-x>* notation stands for *<Control-X>*, that is pressing the *Control* key and the *X* letter key at the same time. The mouse actions can be a *Single* or a *Double* click on the *\*L\*eft*, *\*M\*iddle* or *\*R\*ight* button.



View	Key/Mouse	Action
ControlView	<Return> <Delete>	On a view line, raises this view window On a view line, closes the window and its children windows
TreeView	<Backspace> <Single-L> <Return> <Single-R> <C-c> <C-x> <C-v> <C-y> <C-w> <C-t> <C-a> <C-s> <C-e> <C-b> <C-u> <C-d> <C-l> <C-z>	Raises the ControlView window Select current node On a node, open/close the selected node Opens the pop-up menu for selected node Copy selected node Cut selected node Paste buffered tree as selected node's brother Paste buffered tree as selected node's child Opens a new <i>Tree window</i> for selected node Opens a new <i>Table window</i> for selected node Change selected node's name Change selected node's CGNS type Change selected node's value Add brother node to selected node Add child node to selected node Copy selected node as link destination Add link to buffered link node destination Check selected node and its sub-tree
TableView	<Backspace>	Raises the ControlView window



# PATTERN VIEW



# VTK VIEW

## 5.1 Camera

The camera is your eye, when you move around the object you are looking at the camera is moving, not the object. The object has a position in the 3D space, its coordinates are relative to an origin.

## 5.2 Key bindings

The bindings are the default VTK behaviour, see below how you can interact with the scene depending on the mode you have. The default mode is the *Trackball Camera* mode (note that all other modes are unactivated in this CGNS.NAV version).

t Trackball Camera mode.

mouse left Rotation mouse right Zoom mouse middle Panning ctrl - left Spinning



## QUERY VIEW





## LINK VIEW



## TABLE VIEW

---

**Warning:** There are a *lot* of screenshots in this CGNS .NAV doc, some may be a bit out-dated but most of the look-and-feel of the tool would keep unchanged.



# NAV INDEX

- *genindex*