



pyCGNS.NAV/Manual

Release 4.0.1

Marc Poinot

January 26, 2011

CONTENTS

1	QuickStart	3
2	Tree View	7
3	Option View	9
3.1	Options	9
4	Pattern View	11
5	VTK View	13
5.1	Camera	13
5.2	Key bindings	13
6	Query View	15
7	Link View	17
8	Table View	19

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

```
CGNS.NAV
```

and the pyCGNS browser appears. 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.

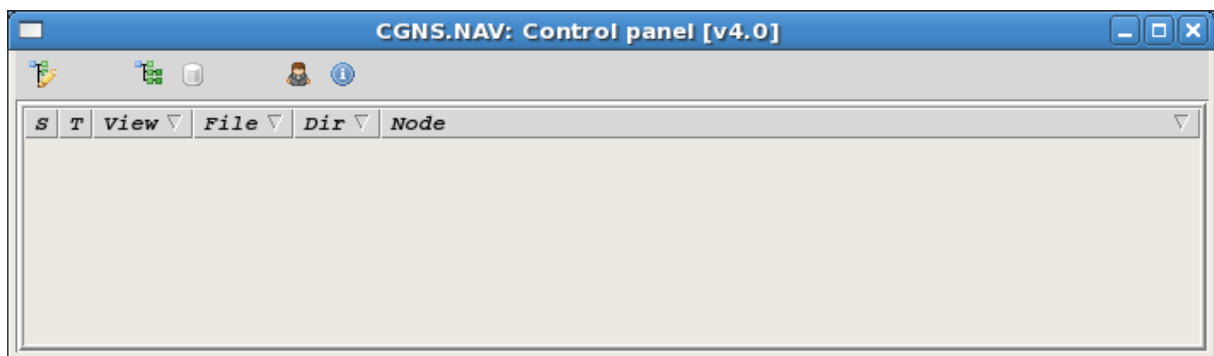
The file format is detected using the file extension. You can give a list of files to `CGNS.NAV`, it opens each file w.r.t. its type:

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

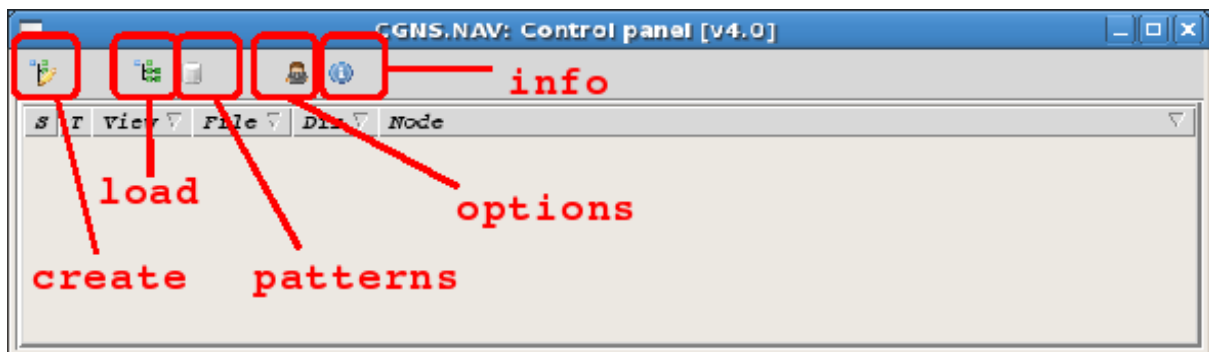
The `.cgns` extension uses the CGNS library which can detect both CGNS/ADF and CGNS/HDF formats.

QUICKSTART

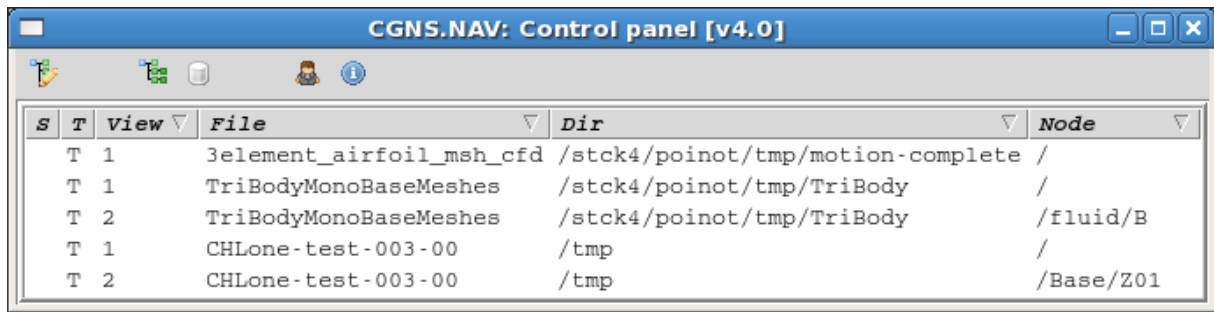
The first window you see when you start `CGNS.NAV` is the *Control View*. This main view helps you to find out the views you have and on which CGNS tree they refer to. If you select the line of an existing view in the *Control view* and press *Return*, this view will be raised on your screen. You can 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.

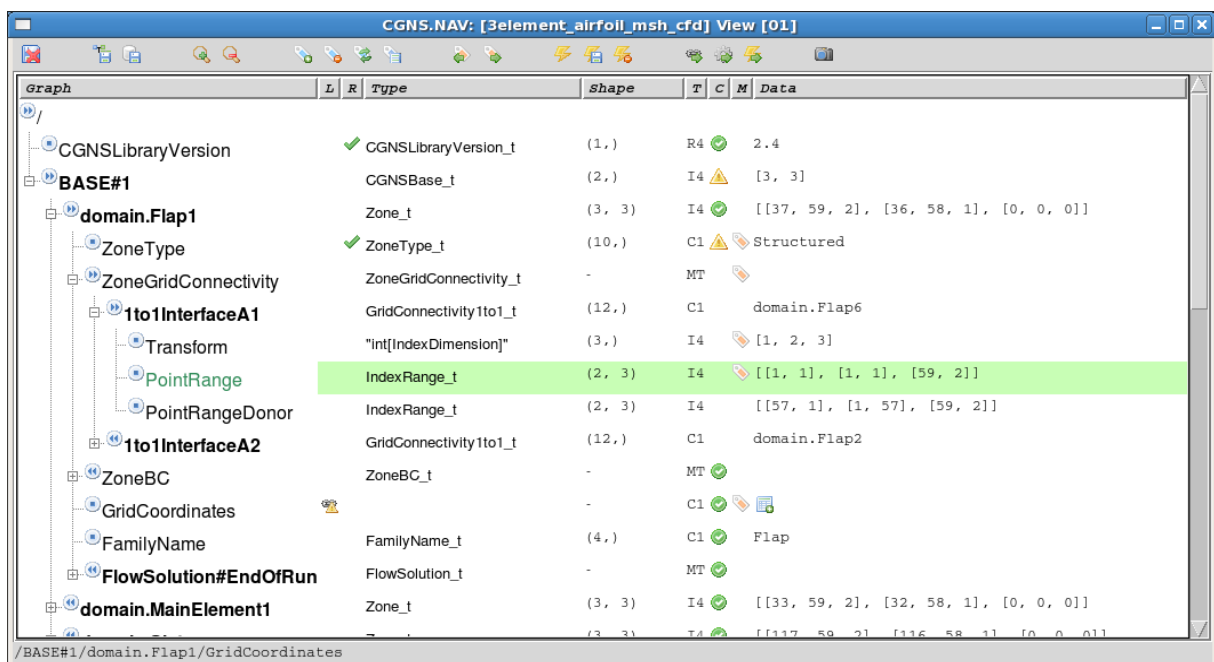


S	T	View	File	Dir	Node
T	1		3element_airfoil_msh_cfd	/stck4/poinot/tmp/motion-complete	/
T	1		TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/
T	2		TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/fluid/B
T	1		CHLone-test-003-00	/tmp	/
T	2		CHLone-test-003-00	/tmp	/Base/Z01

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.



Graph	L	R	Type	shape	T	C	M	Data
/								
CGNSLibraryVersion	✓		CGNSLibraryVersion_t	(1,)	R4	✓		2.4
BASE#1			CGNSBase_t	(2,)	I4	⚠		[3, 3]
domain.Flapp1			Zone_t	(3, 3)	I4	✓		[[37, 59, 2], [36, 58, 1], [0, 0, 0]]
ZoneType	✓		ZoneType_t	(10,)	C1	⚠		Structured
ZoneGridConnectivity			ZoneGridConnectivity_t	-	MT			
1to1InterfaceA1			GridConnectivity1to1_t	(12,)	C1			domain.Flapp6
Transform			"int[IndexDimension]"	(3,)	I4			[1, 2, 3]
PointRange			IndexRange_t	(2, 3)	I4	⚠		[[1, 1], [1, 1], [59, 2]]
PointRangeDonor			IndexRange_t	(2, 3)	I4			[[57, 1], [1, 57], [59, 2]]
1to1InterfaceA2			GridConnectivity1to1_t	(12,)	C1			domain.Flapp2
ZoneBC			ZoneBC_t	-	MT	✓		
GridCoordinates				-	C1	✓		
FamilyName			FamilyName_t	(4,)	C1	✓		Flap
FlowSolution#EndOfRun			FlowSolution_t	-	MT	✓		
domain.MainElement1			Zone_t	(3, 3)	I4	✓		[[33, 59, 2], [32, 58, 1], [0, 0, 0]]
				(3, 3)	I4	✓		[[117, 59, 2], [116, 58, 1], [0, 0, 0]]

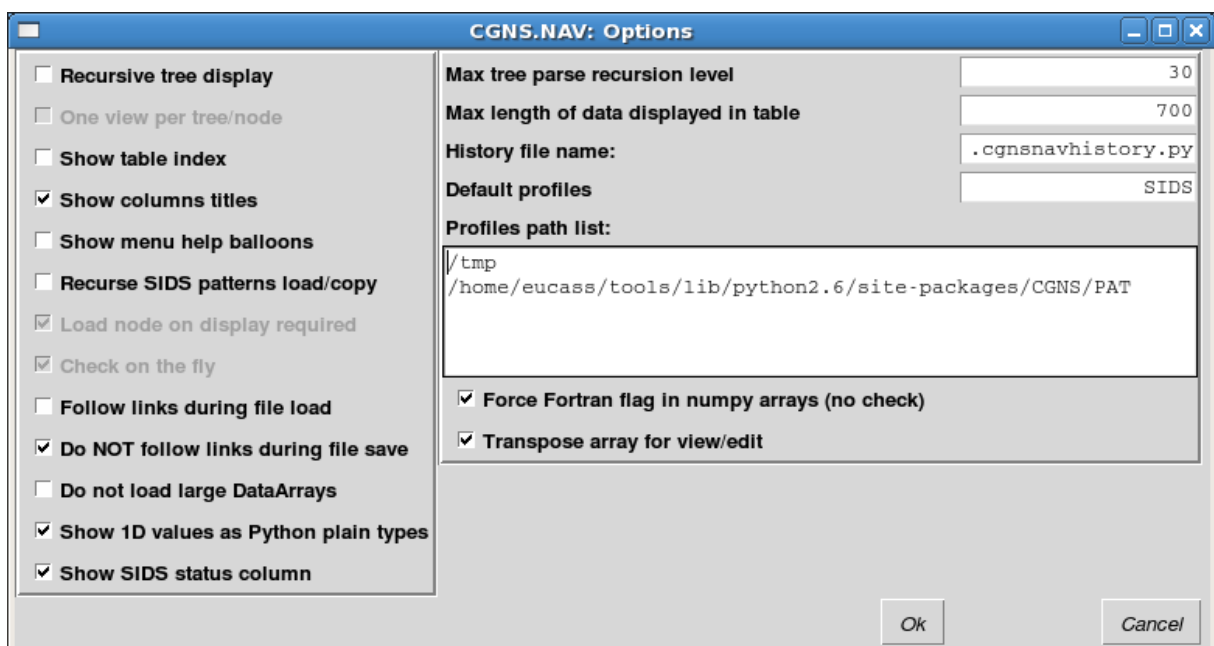
You click on the +/- icon before any node name in the tree to open/close the child sub-tree for this node. A mouse left button simple click on a node selects the node. A mouse right button simple click opens the menu for the selected node. The menu gives you access to some function such as Copy/Paste, 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, 2], [3
ZoneType	Change name (C-a)	(10,)	C1	Structured
ZoneGridConnectivity	Change CGNS type	-	MT	
1to1InterfaceA1	Change CGNS type (C-s)	(12,)	C1	domain.Flap6
Transform	Change data type	(3,)	I4	[1, 2, 3]
PointRange	Change value (C-e)	(2, 3)	I4	[[1, 1], [1, 1],
PointRangeDonor	Add link (C-l)	(2, 3)	I4	[[57, 1], [1, 57
1to1InterfaceA2	Remove link	(12,)	C1	domain.Flap2
ZoneBC	Copy (C-c)	-	MT	
GridCoordinates	Cut (C-x)	(4,)	C1	Flap
FamilyName	Paste as brother (C-v)	-	MT	
FlowSolution#EndOfRun	Paste as child (C-y)	(3, 3)	I4	[[33, 59, 2], [3
domain.MainElement1	Open tree view (C-w)	(3, 3)	I4	[[117, 59, 2], [1
domain.Slat	Open table view (C-t)	(3, 3)	I4	[[29, 59, 2], [2
domain.Flap2	Add brother (C-b)	(3, 3)	I4	[[15, 59, 2], [1
domain.Flap3	Add child (C-u)	(3, 3)	I4	[[25, 59, 2], [2
domain.Flap4	Check (C-z)	(3, 3)	I4	[[57, 59, 2], [5
domain.Flap5	Zone_t	(3, 3)	I4	[[55, 53, 2], [5
domain.Flap6				
domain.MainElement2				

TREE VIEW

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.



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

ControlView	<Return>	On a view line, raises this view window
TreeView	<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

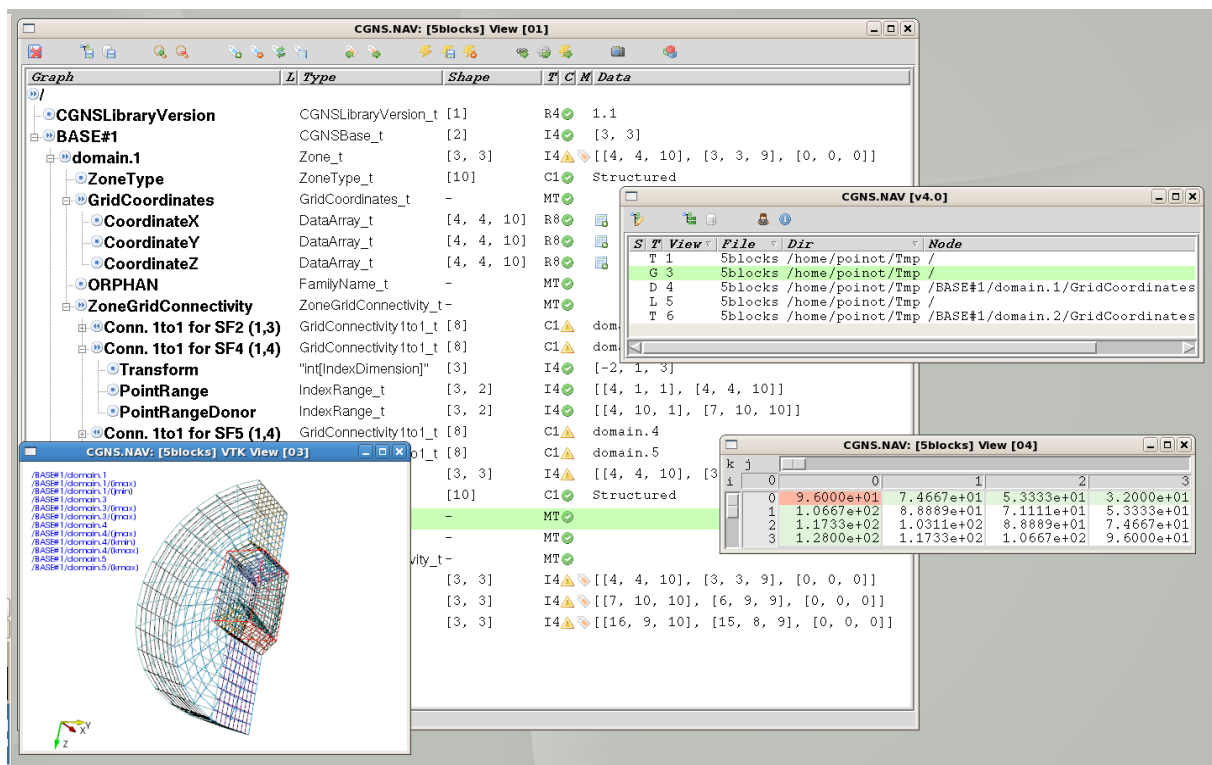
QUERY VIEW

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.

The screenshot below shows a CGNS .NAV with various windows.



- *genindex*