pyCGNS.NAV/Manual

Release 4.2.0

Marc Poinot

October 19, 2012

CONTENTS

The CGNS.NAV tool is a CGNS tree browser.

There are three important views (or windows) you can open on a CGNS tree. The `control view` is the manager for all the views on all the trees you would open. The `tree view` is the main window you can open on a single tree. The `VTK view` displays the mesh, element sets, connectivities and boundary conditions of your CGNS base.

Note: The screenshots you have on these pages may be outdated. Most features are unchanged from GUI version to another GUI version, but as the update of all screenshots is long and tedious the actual GUI you would have on your screen may be more or less close to these docs...

CHAPTER

ONE

CONTROL VIEW

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

```
CGNS.NAV
```

`CGNS.NAV` *options*

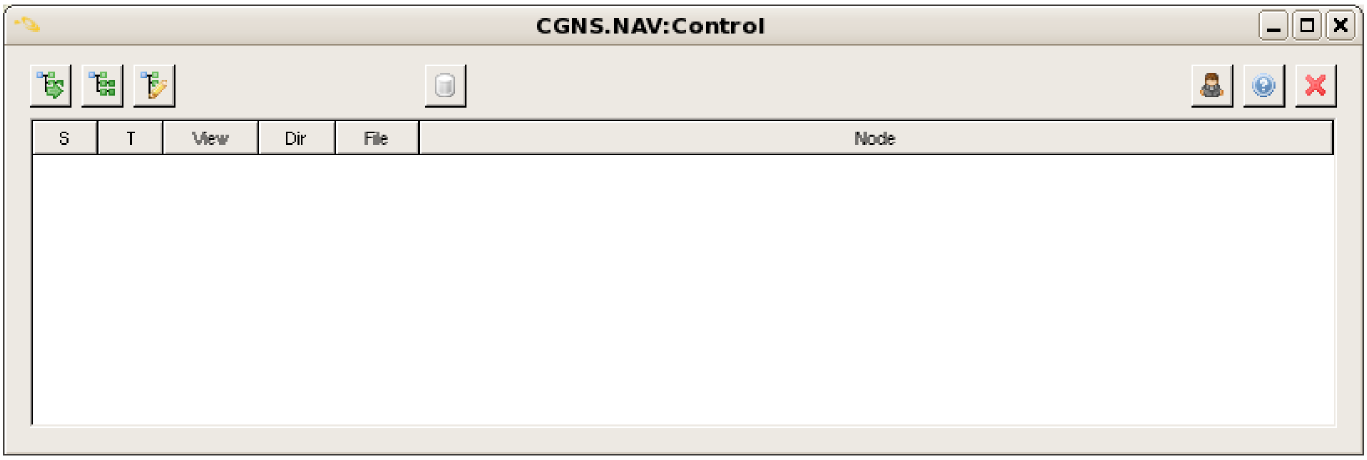
You can use several options to launch the `CGNS.NAV` program.

If you type `CGNS.NAV -R file.cgns` , the `control view` containing the file.cgns file is opened and the `tree view` of the file is also opened. When the `-R` option is used, all the nodes of the tree displayed in the `tree view` are expanded. If you don't enter a file name after the `-R` option, only the `control view` without loaded CGNS file appears. But when you choose your CGNS file, you obtain a tree whose all the nodes are expanded in the `tree view` which is then opened.

If you type `CGNS.NAV -l` , you launch the `control view` and the `tree view` of the last used CGNS file. In the `tree view`, the nodes are not expanded unlike the previous `-R` option. If you enter a file name after the `-l` option, you open two additional views, namely the `control view` and the `tree view` of the loaded file.

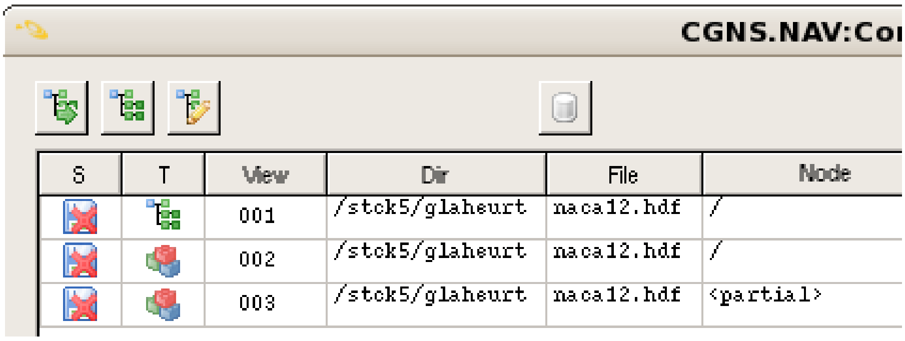
You can use the `-g` option by entering `CGNS.NAV -g file.cgns` . The `control view` , the `tree view` and the `VTK view` of the file.cgns file are displayed on the screen. In the `tree view` , the nodes of the tree aren't expanded. In the `control view` are listed the two views, namely the `tree view` and the `VTK view` . If you don't type a file name after the `-g` option, only the empty `control view` is opened. The two anothers views will be launched, when you will select the CGNS filed to load in the `control view`.

After the program's launching, the `control view` appears:



This view helps you to find out the views you deal with and to which CGNS tree they correspond.

You have an example of several views listed in the `control view` below:


CGNS.NAV:Co

S	T	View	Dir	File	Node
☒		001	/stck5/glaheurt	naca12.hdf	/
☒		002	/stck5/glaheurt	naca12.hdf	/
☒		003	/stck5/glaheurt	naca12.hdf	<partial>

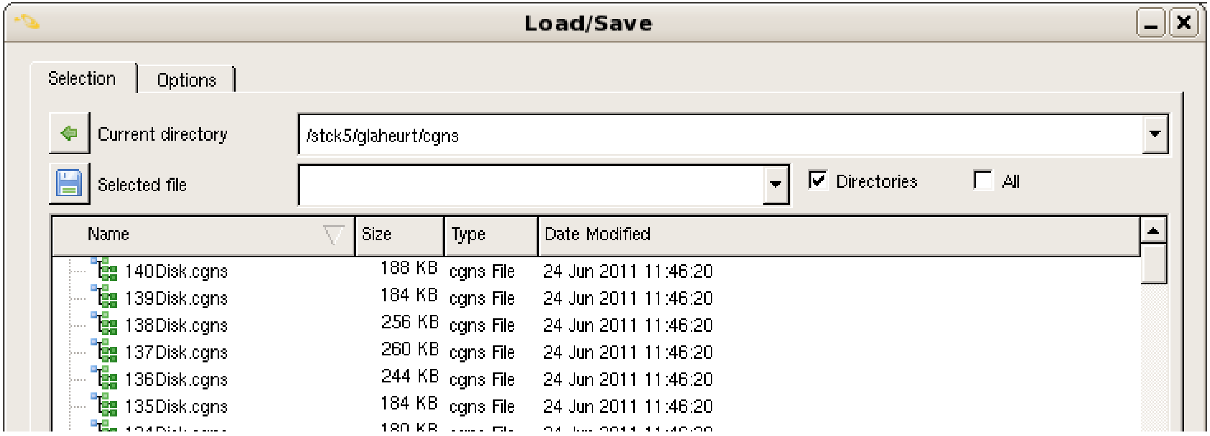
If you select the line of an existing view, this view will be raised on your screen. You can find back the `control view` from any other view just by clicking the 🏠 button located in the bottom left-hand corner of the view.

Each time you open a view, the `control view` keeps track of it.

The columns of the lines which you see in this `control view` are:

Column	Definition
`S`	Status flag, says if the tree has been modified and needs a save.
`T`	Type of view. The `tree views` are marked with , the `VTK view` with , the `form view` with and the `queries view` with .
`View`	The number of views which you opened.
`Dir`	The directory where you loaded the file.
`File`	The file name.
`Node`	Status flag, says if the entire tree is displayed `/` or if only a part of the tree is visualized `<partial>`.

If you click the button, this window appears:


Load/Save

Current directory: /stck5/glaheurt/cgns

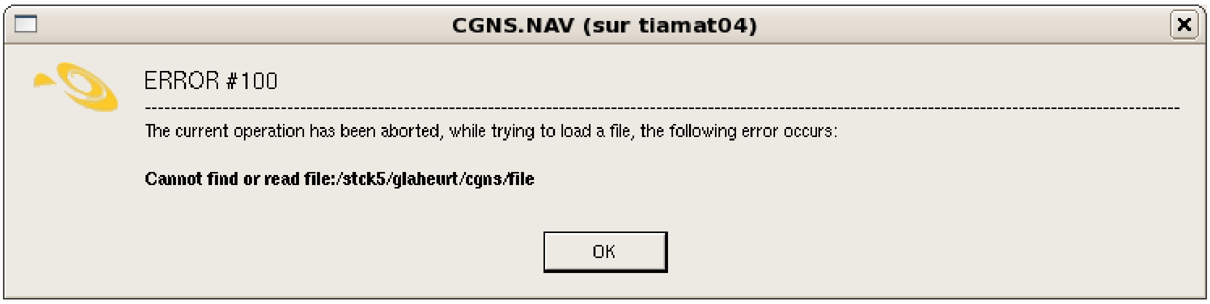
Selected file:

☑ Directories ☐ All

Name	Size	Type	Date Modified
140Disk.cgns	188 KB	cgns File	24 Jun 2011 11:46:20
139Disk.cgns	184 KB	cgns File	24 Jun 2011 11:46:20
138Disk.cgns	256 KB	cgns File	24 Jun 2011 11:46:20
137Disk.cgns	260 KB	cgns File	24 Jun 2011 11:46:20
136Disk.cgns	244 KB	cgns File	24 Jun 2011 11:46:20
135Disk.cgns	184 KB	cgns File	24 Jun 2011 11:46:20
134Disk.cgns	180 KB	cgns File	24 Jun 2011 11:46:20

Know, you must select an existing CGNS file by typing its directory's path in the box located to the right of the Current directory icon and by entering its name in the box located to the right of the Selected file icon. You must click the Selected file button to valid your choice.

If you choose an unexisting file name, the following error message is returned to you:



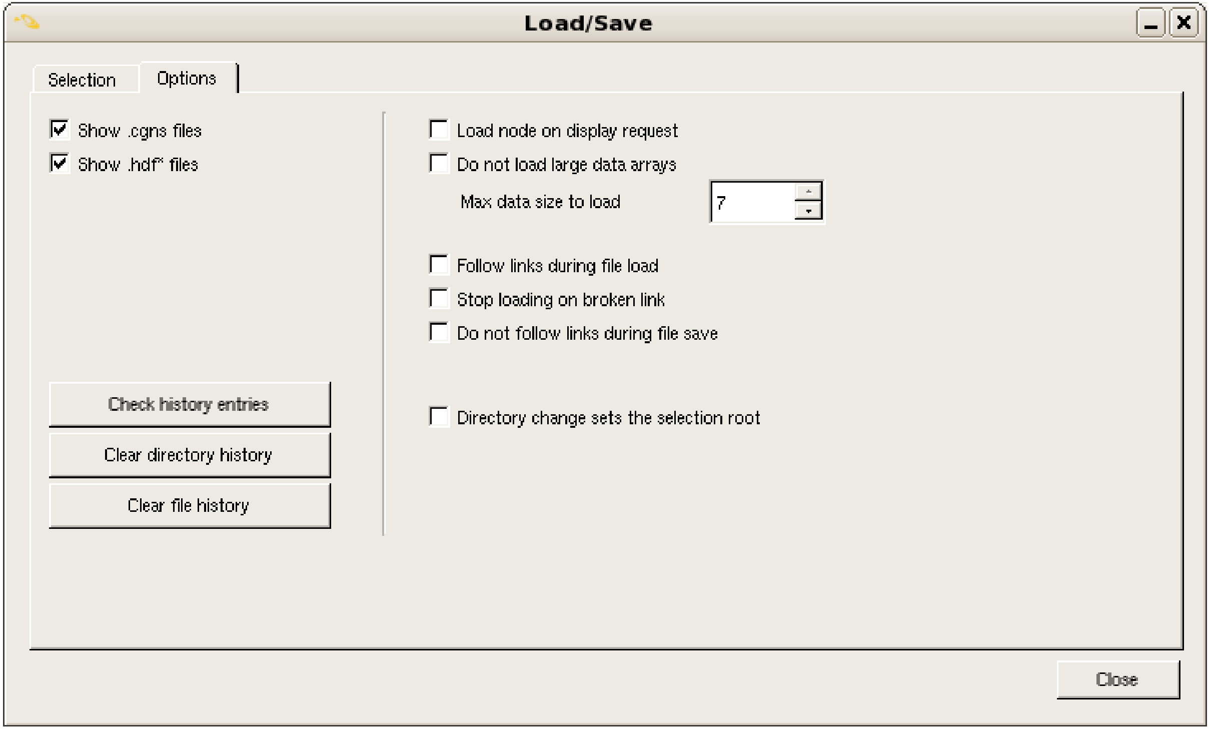
CGNS.NAV (sur tiamat04)

ERROR #100

The current operation has been aborted, while trying to load a file, the following error occurs:

Cannot find or read file:/stck5/glaheurt/cgns/file

The options of the `Load/Save` window are shown below:



Load/Save

☑ Show .cgns files

☑ Show .hdf* files

□ Load node on display request

□ Do not load large data arrays

Max data size to load 7

□ Follow links during file load

□ Stop loading on broken link

□ Do not follow links during file save

□ Directory change sets the selection root

To deal with a CGNS file

Icon	Action
	Load a new CGNS file.
	Load the last used CGNS file.
	Create a new CGNS file from scratch.
🛢	Open the CGNS/SIDS sub-trees database.
👤	Set the user defined options.
❓	About.
✖	Close all the CGNS.NAV windows.

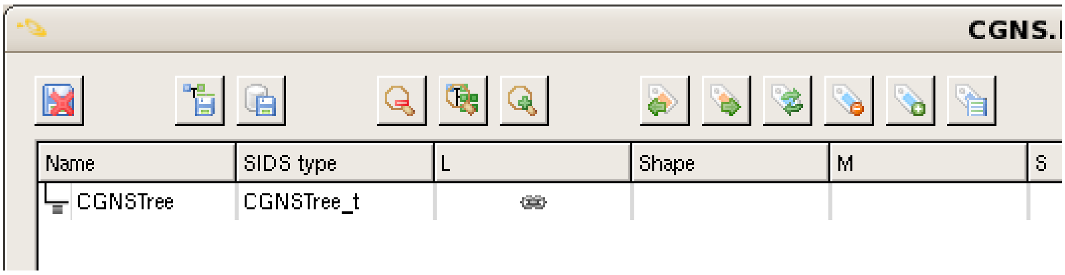
You can also reload the last used CGNS file by clicking on the icon .

CHAPTER

TWO

TREE VIEW

When you open a new CGNS file or if you load again the last used CGNS file, the tree view appears:

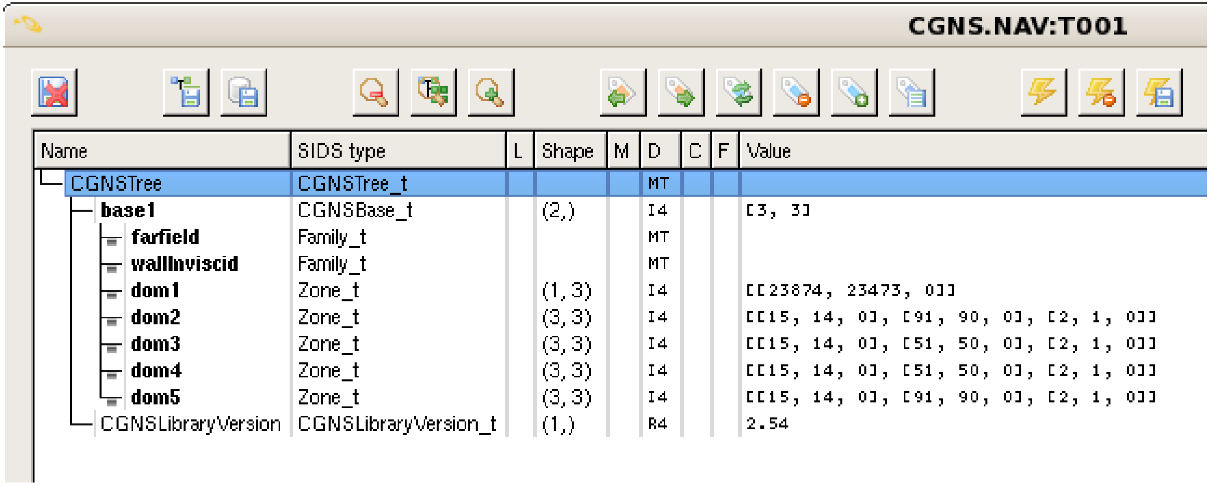


Name	SIDS type	L	Shape	M	S
CGNSTree	CGNSTree_t				

The tree view displays the nodes for the loaded CGNS file. It is the main view of the tree. You can browse or perform modifications of the loaded CGNS tree in this view.

As you can observe it, there is only one entry in the tree view. This is the root of our CGNS file which can contain one or several bases. All the nodes are arranged in a tree structure. The opening of the CGNS tree occurs in a recursive way. A node is selected by clicking the mouse Button 1 on the wanted node. The path of the selected node is displayed in the box at the bottom of the window. If you want to expand this node one level up in order to display the entries corresponding to the base contained in the CGNSTree node, click the icon.

If you perform this operation again with the base1 node, you obtain the following tree view:



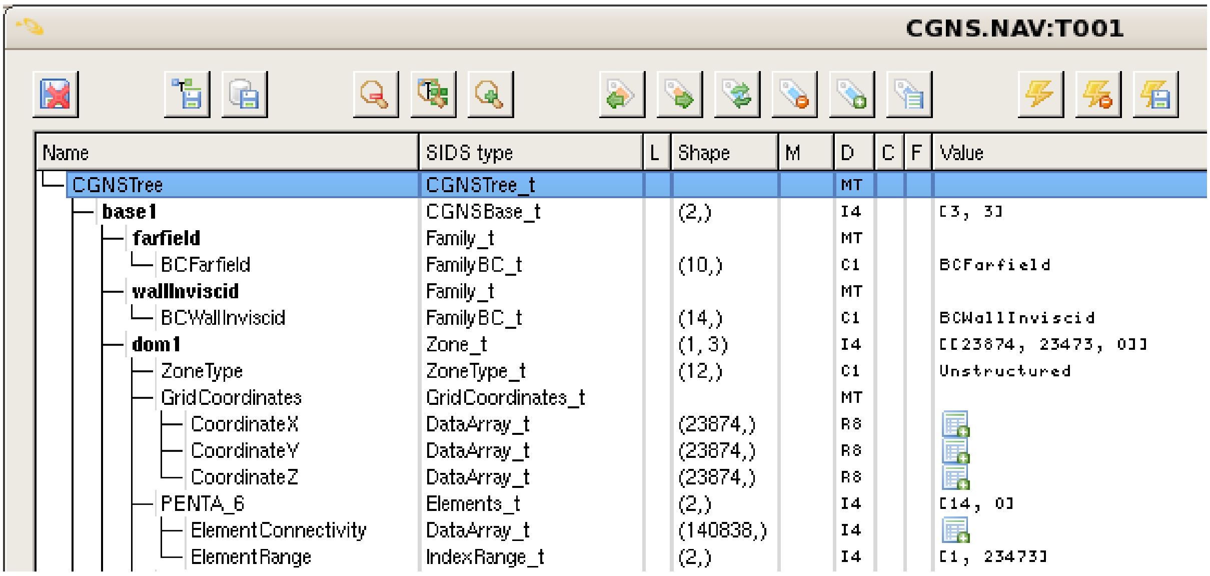
Name	SIDS type	L	Shape	M	D	C	F	Value
CGNSTree	CGNSTree_t				MT			
base1	CGNSBase_t		(2,)		I4			[3, 3]
farfield	Family_t				MT			
wallInviscid	Family_t				MT			
dom1	Zone_t		(1, 3)		I4			[[23874, 23473, 0]]
dom2	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
dom3	Zone_t		(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom4	Zone_t		(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom5	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
CGNSLibraryVersion	CGNSLibraryVersion_t		(1,)		R4			2.54

You can repeat the operation for the nodes of the different zones which are under the base1 node. And so on...

In order to expand the tree view one level down, click the icon.

To expand all the loaded CGNS/tree, you must click .

You can see the expanded CGNS tree shown below:

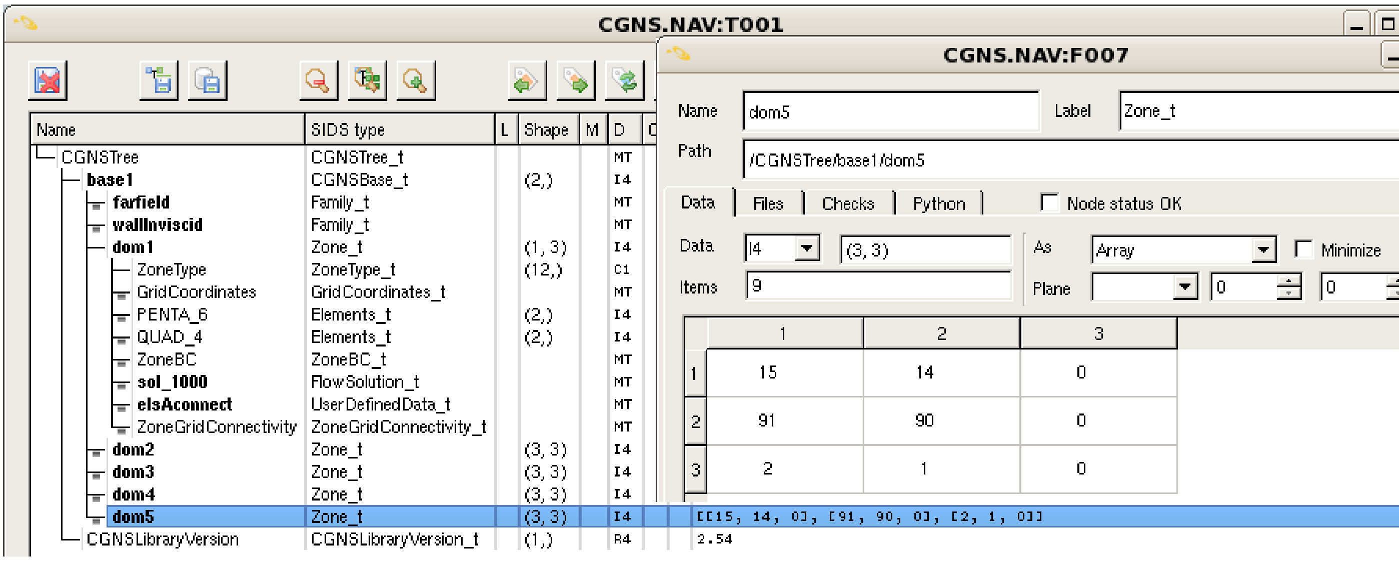


Name	SIDS type	L	Shape	M	D	C	F	Value
CGNSTree	CGNSTree_t				MT			
base1	CGNSBase_t		(2,)		I4			[3, 3]
farfield	Family_t				MT			
BCFarfield	FamilyBC_t		(10,)		C1			BCFarfield
wallInviscid	Family_t				MT			
BCWallInviscid	FamilyBC_t		(14,)		C1			BCWallInviscid
dom1	Zone_t		(1, 3)		I4			[[23874, 23473, 0]]
ZoneType	ZoneType_t		(12,)		C1			Unstructured
GridCoordinates	GridCoordinates_t				MT			
CoordinateX	DataArray_t		(23874,)		R8			
CoordinateY	DataArray_t		(23874,)		R8			
CoordinateZ	DataArray_t		(23874,)		R8			
PENTA_6	Elements_t		(2,)		I4			[14, 0]
ElementConnectivity	DataArray_t		(140838,)		I4			
ElementRange	IndexRange_t		(2,)		I4			[1, 23473]

If you want to open the child sub-tree of the selected node, you click on the ▾ icon located before any node. When you perform this operation, this ▾ icon disappears. You can collapse all the nodes below a node by clicking near to the name of the desired node at the place where was the ▾ icon.

When you use the 🔍 icon, you expand all the nodes one level up while when you click on the ▾ icon, you expand only the wanted node one level up.

If you click on the ☰ icon, the `form view` of the selected node appears on the screen:



Name	SIDS type	L	Shape	M	D	Value
CGNSTree	CGNSTree_t				MT	
base1	CGNSBase_t		(2,)		I4	
farfield	Family_t				MT	
wallInviscid	Family_t				MT	
dom1	Zone_t		(1, 3)		I4	
ZoneType	ZoneType_t		(12,)		C1	
GridCoordinates	GridCoordinates_t				MT	
PENTA_6	Elements_t		(2,)		I4	
QUAD_4	Elements_t		(2,)		I4	
ZoneBC	ZoneBC_t				MT	
sol_1000	FlowSolution_t				MT	
elsAconnect	UserDefinedData_t				MT	
ZoneGridConnectivity	ZoneGridConnectivity_t				MT	
dom2	Zone_t		(3, 3)		I4	
dom3	Zone_t		(3, 3)		I4	
dom4	Zone_t		(3, 3)		I4	
dom5	Zone_t		(3, 3)		I4	[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
CGNSLibraryVersion	CGNSLibraryVersion_t		(1,)		R4	2.54

Name: dom5 Label: Zone_t

Path: /CGNSTree/base1/dom5

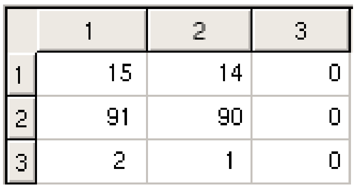
Data | Files | Checks | Python □ Node status OK

Data: I4 (3, 3) As: Array □ Minimize

Items: 9 Plane: 0 0

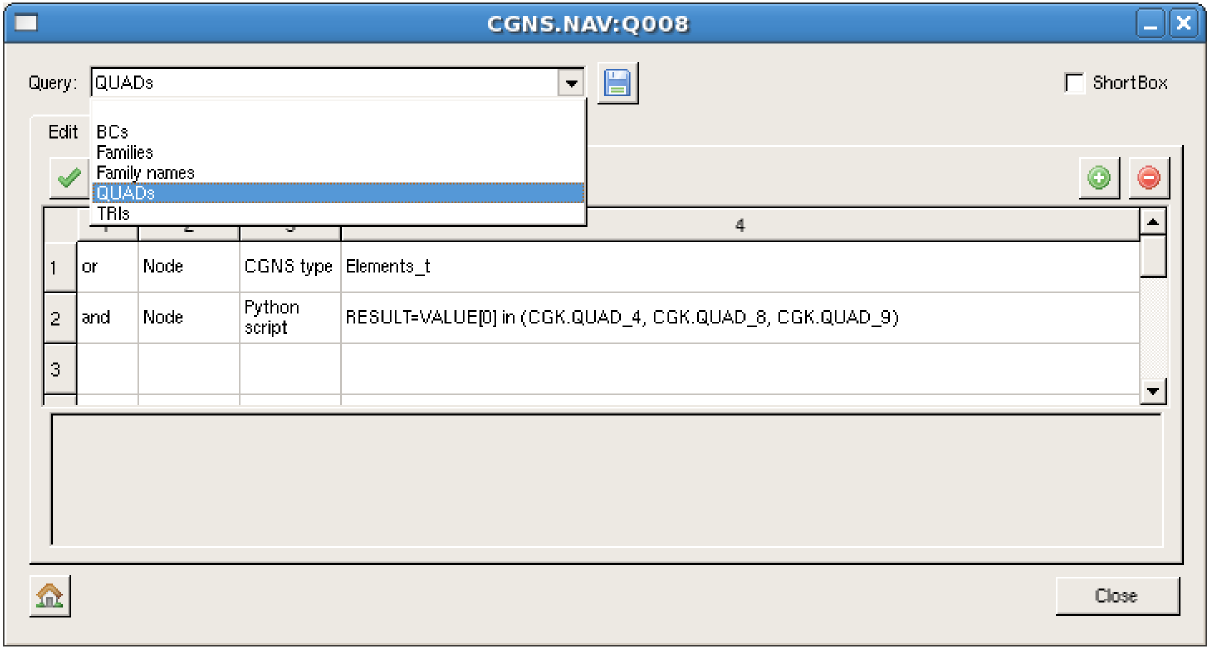
	1	2	3
1	15	14	0
2	91	90	0
3	2	1	0

As you can observe it, the node name, the node type, the node path, the node data, the node shape and the values defined in the selected node are displayed in the `form view` . You can also resize the table by checking ☑ Minimize the case. The table's dimensions are adjusted so that the column's length is equal to the string's length displayed, like below:



	1	2	3
1	15	14	0
2	91	90	0
3	2	1	0

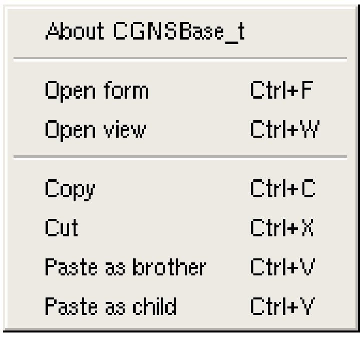
You can open the `queries view` by clicking on the icon and this view appears:



	1	2	3	4
1	or	Node	CGNS type	Elements_t
2	and	Node	Python script	RESULT=VALUE[0] in (CGK.QUAD_4, CGK.QUAD_8, CGK.QUAD_9)
3				

Modify the CGNS tree

A mouse right button simple click on a selected node allows you to open the popup menu displayed below:

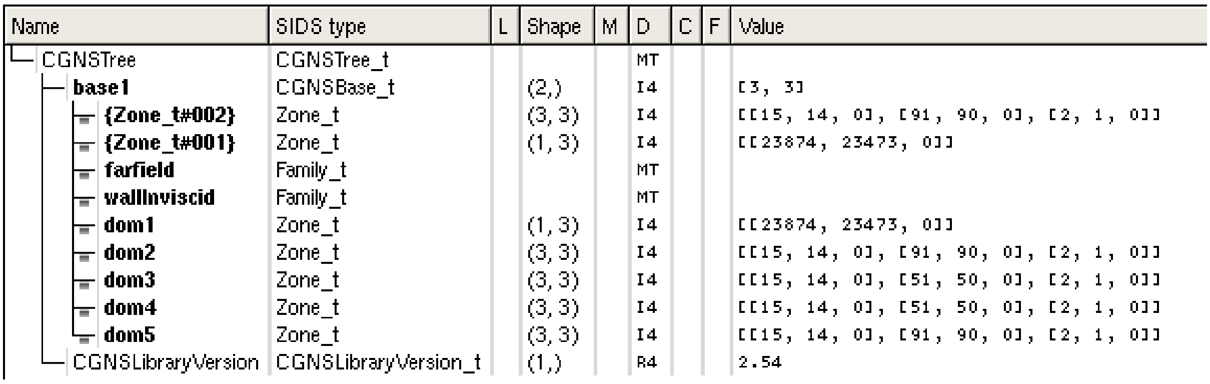


This menu gives you access to some function such as `Open a view` and `Copy` , `Cut` , `Paste` a node. These functions have keyboards shortcuts listed below.

Key bindings

Button	Action
`Ctrl` + `F`	Open the form.
`Ctrl` + `W`	Open a new view of the tree.
`Ctrl` + `C`	Copy the selected node.
`Ctrl` + `X`	Cut the selected node.
`Ctrl` + `V`	Paste the copied node as brother node of the selected node.
`Ctrl` + `Y`	Paste the copied node as child node of the selected node.

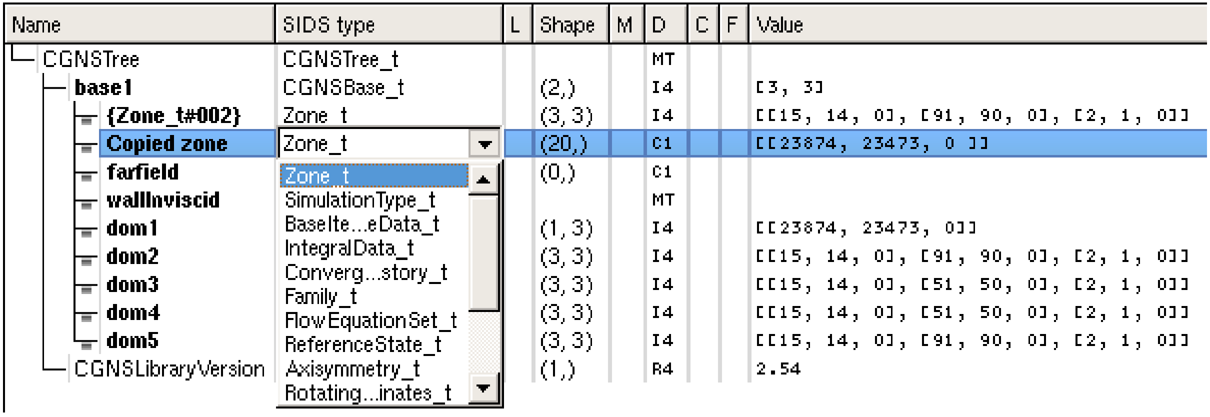
The `tree view` allows to perform modifications in the CGNS trees. It supports the `Copy`, `Cut`, `Paste` functions. You can copy a sub-tree of a view and paste it into the same view or into another view which can belong to a different CGNS tree. You press `Ctrl` + `C` on the selected node or you choose `Copy` in the popup menu opened by a mouse right button simple click on the wanted node to copy it into a buffer. You paste the copied node by pressing `Ctrl` + `V` or by selecting `Paste as brother` in the popup menu. The new node becomes the brother node of the selected node. By contrast, if you select `Paste as child` in the popup menu, the new node becomes the child node of the selected node.



Name	SIDS type	L	Shape	M	D	C	F	Value
CGNSTree	CGNSTree_t				MT			
base1	CGNSBase_t		(2,)		I4			[3, 3]
{Zone_t#002}	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
{Zone_t#001}	Zone_t		(1, 3)		I4			[[23874, 23473, 0]]
farfield	Family_t				MT			
wallInviscid	Family_t				MT			
dom1	Zone_t		(1, 3)		I4			[[23874, 23473, 0]]
dom2	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
dom3	Zone_t		(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom4	Zone_t		(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom5	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
CGNSLibraryVersion	CGNSLibraryVersion_t		(1,)		R4			2.54

In this example, the `dom1` and `dom2` nodes are copied and pasted into the same view. Here, their copies are automatically named `{Zone_t#002}` and `{Zone_t#001}` . Indeed, these copied nodes have the same content as their origin nodes.

You can modify these new nodes by a double left-click on their names, their types and their values. Type then just the new name and the new value. You must select the new `SIDS type` from the drop-down list opened by clicking in the entry field to change the node type, like below:



Name	SIDS type	L	Shape	M	D	C	F	Value
CGNSTree	CGNSTree_t				MT			
base1	CGNSBase_t		(2,)		I4			[3, 3]
{Zone_t#002}	Zone_t		(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
Copied zone	Zone_t		(20,)		C1			[[23874, 23473, 0]]
farfield			(0,)		C1			
wallInviscid					MT			
dom1			(1, 3)		I4			[[23874, 23473, 0]]
dom2			(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
dom3			(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom4			(3, 3)		I4			[[15, 14, 0], [51, 50, 0], [2, 1, 0]]
dom5			(3, 3)		I4			[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
CGNSLibraryVersion			(1,)		R4			2.54

The `Cut` function acts as the `Copy` function with the difference that the selected node is removed from the `tree view`.

To deal with the CGNS tree

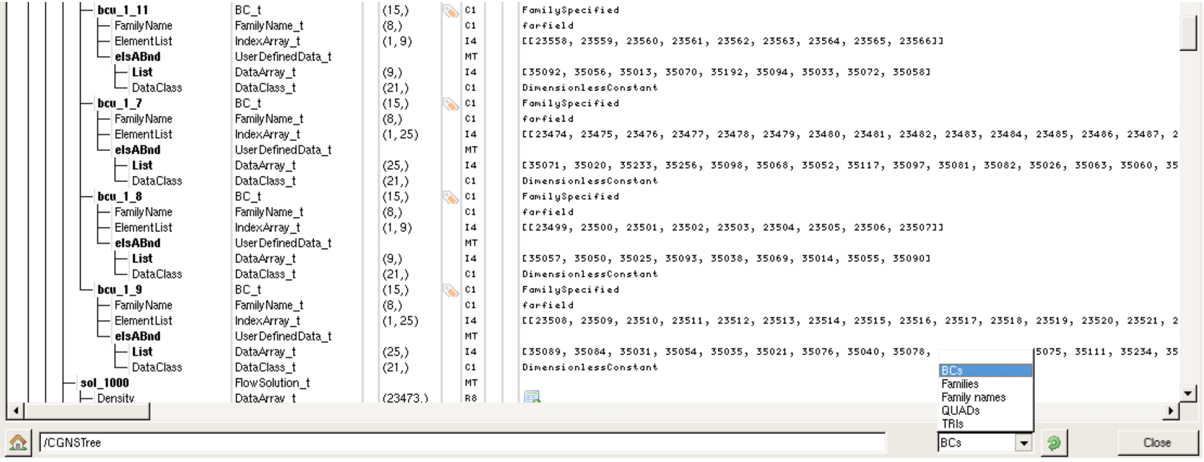
Icon	Action
	Expand the tree one level down.
	Expand all the tree.
	Expand the tree one level up.
	Select the previous marked node.
	Select the next marked node.
	Unselect the marked nodes and flag the unmarked nodes.
	Unmark all the nodes.
	Mark all the nodes.
	Open the selected nodes list.
	Open the queries window.
	Open the form view of the selected node.
	Display the mesh of the tree in the `VTK view`.

Mouse bindings

The mouse bindings and the corresponding actions are:

Button	Action
`Button 1`	Select a node by a simple click on the wanted node.
`Button 1`	Expand/Collapse all the nodes below the selected node by clicking on the ≡ icon.
`Button 1`	Modify the name of the selected node, its `SIDS type` or its value by a double-click on the desired column of the node.

The `tree view` allows you to choose all the elements which have the same `SIDS type`. For example, if you want to select all the elements whose `SIDS type` is `BC_t`, select this type in the drop-down list located in the bottom right-hand corner of the `tree view` and click the ⟳ icon in order to apply the selection.



bcu_1_11	BC_t	(15,)	🏷	C1	FamilySpecified
FamilyName	FamilyName_t	(8,)		C1	farfield
ElementList	IndexArray_t	(1, 9)		I4	[[23558, 23559, 23560, 23561, 23562, 23563, 23564, 23565, 23566]]
elsABnd	UserDefinedData_t			MT	
List	DataArray_t	(9,)		I4	[35092, 35056, 35013, 35070, 35192, 35094, 35033, 35072, 35058]
DataClass	DataClass_t	(21,)		C1	DimensionlessConstant
bcu_1_7	BC_t	(15,)	🏷	C1	FamilySpecified
FamilyName	FamilyName_t	(8,)		C1	farfield
ElementList	IndexArray_t	(1, 25)		I4	[[23474, 23475, 23476, 23477, 23478, 23479, 23480, 23481, 23482, 23483, 23484, 23485, 23486, 23487, 2
elsABnd	UserDefinedData_t			MT	
List	DataArray_t	(25,)		I4	[35071, 35020, 35233, 35256, 35098, 35068, 35052, 35117, 35097, 35081, 35082, 35026, 35063, 35060, 35
DataClass	DataClass_t	(21,)		C1	DimensionlessConstant
bcu_1_8	BC_t	(15,)	🏷	C1	FamilySpecified
FamilyName	FamilyName_t	(8,)		C1	farfield
ElementList	IndexArray_t	(1, 9)		I4	[[23499, 23500, 23501, 23502, 23503, 23504, 23505, 23506, 23507]]
elsABnd	UserDefinedData_t			MT	
List	DataArray_t	(9,)		I4	[35057, 35050, 35025, 35093, 35038, 35069, 35014, 35055, 35090]
DataClass	DataClass_t	(21,)		C1	DimensionlessConstant
bcu_1_9	BC_t	(15,)	🏷	C1	FamilySpecified
FamilyName	FamilyName_t	(8,)		C1	farfield
ElementList	IndexArray_t	(1, 25)		I4	[[23508, 23509, 23510, 23511, 23512, 23513, 23514, 23515, 23516, 23517, 23518, 23519, 23520, 23521, 2
elsABnd	UserDefinedData_t			MT	
List	DataArray_t	(25,)		I4	[35089, 35084, 35031, 35054, 35035, 35021, 35076, 35040, 35078, 5075, 35111, 35234, 35
DataClass	DataClass_t	(21,)		C1	DimensionlessConstant
sol_1000	FlowSolution_t			MT	
Density	DataArray_t	(23473,)		R8	

/CGNSTree

The `tree view` allows you to browse all the marked nodes by using the icon to select the next marked node and the icon to select the previous marked node.

Click the icon while to mark all the nodes, click the icon to unselect all the nodes.

You can also invert the selection by using the button. The marked nodes become unselected and vice versa for the unmarked nodes.

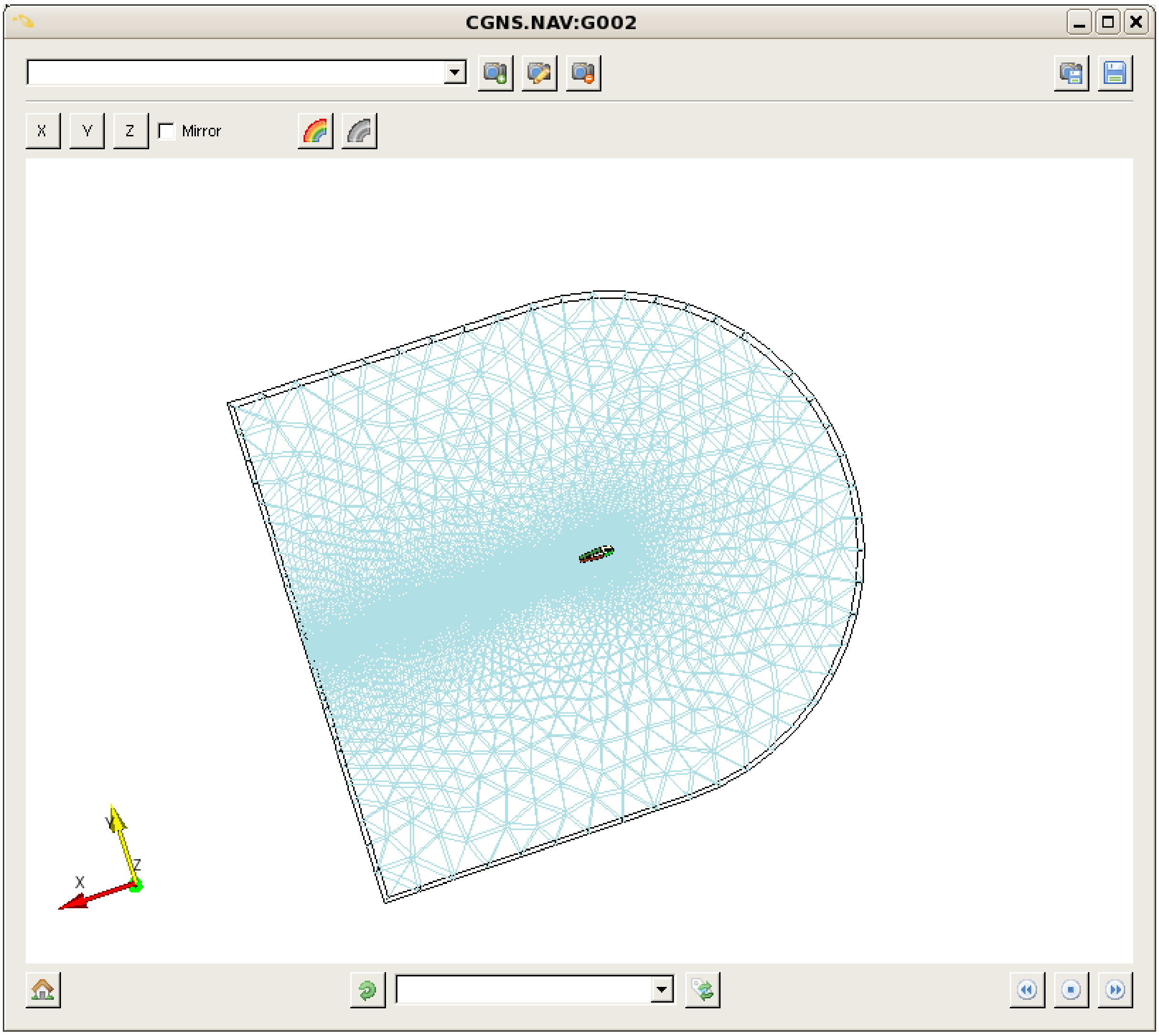
CHAPTER

THREE

VTK VIEW

To display the mesh, element sets, connectivities and boundary conditions contained in the CGNS file, click on .

and the mesh is dispayed in the `VTK view`:



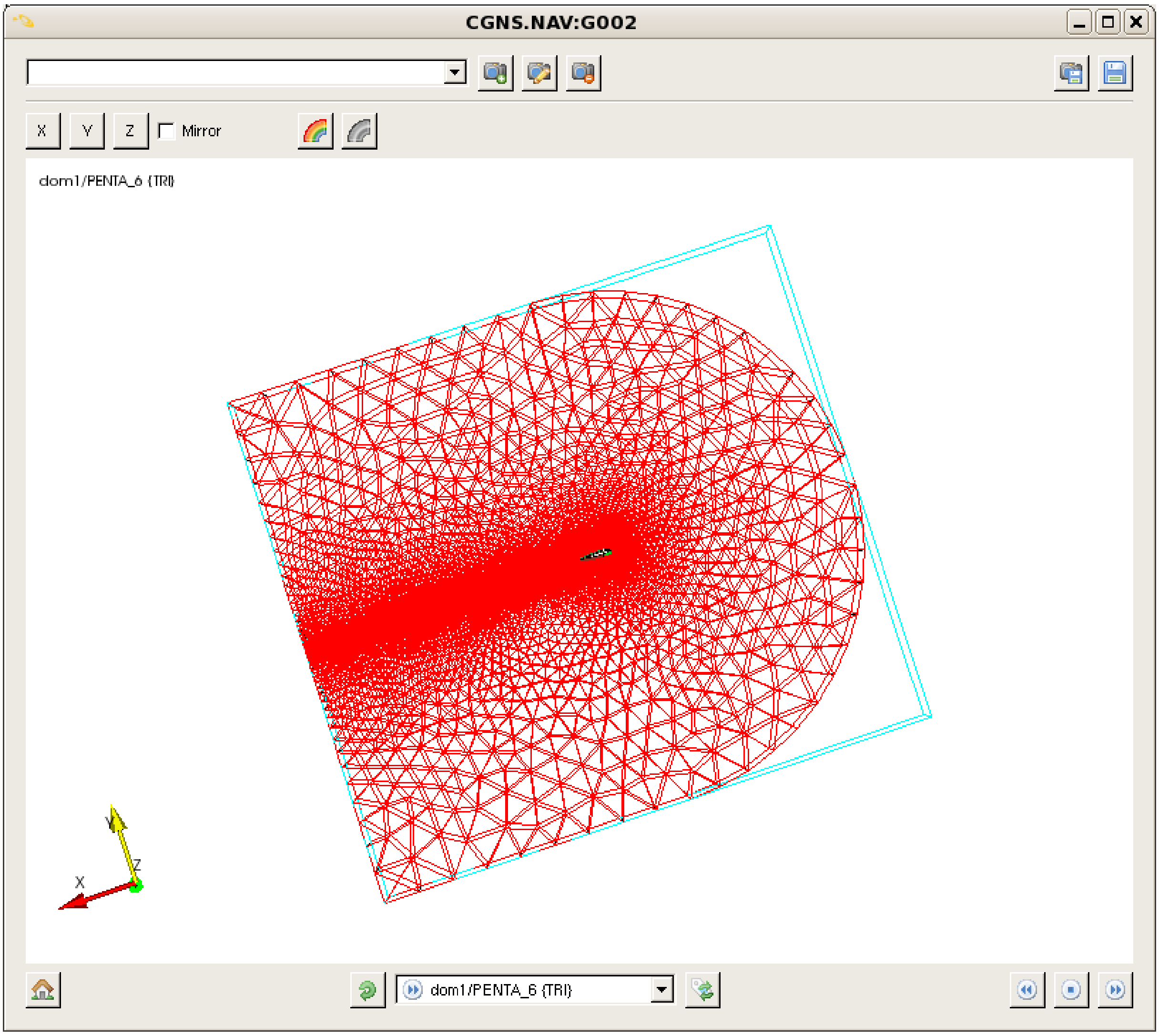
The view can be translated, rotated and scaled by using the mouse. The three axis x,y,z are displayed in the bottom left-hand corner of the window. The x-axis is coloured in red, the y-axis in yellow and the z-axis in green.

Mouse Bindings

The mouse bindings and the related actions are:

Button	Action
`Button 1`	Rotate the camera around its focal point.
`Button 2`	Translate the elements displayed in the window.
`Button 3`	Adjust the view by holding down this button while moving the mouse in the display window. The objects are scaled up when the mouse moves from bottom to up and they are scaled down when the mouse moves from up to bottom.

To select an element of the CGNS/tree, you perform a pick operation by positioning the mouse cursor on the place of your choice and by pressing on the `p` key.

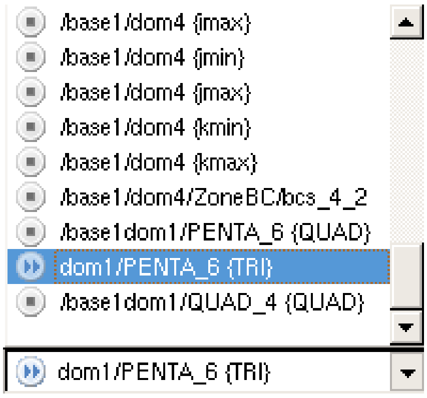


Key Bindings

The following keys and the corresponding actions are:

Key	Action
`f`	Fly to the picked point.
`p`	Perform a pick operation.
`r`	The elements are centered and the camera moves along the current view direction so that all elements are visible in the window .
`s`	Modify the representation of all elements so that they are surfaces.
`w`	Modify the representation of all elements so that they are wireframes.
`d`	Hide the current element selected by performing a pick operation.
`Ctrl`	Add the elements selected by a pick operation to the previous selection.

The pick operation shoots a ray into the 3D scene and returns information about the objects that the ray hits. The first element hit by the ray is highlighted in red and a blue wireframe outlines the bounding box of the selected object.In the top left-hand corner of the `VTK view`, the list of the paths of elements hit by the ray appears. In our case, there is only one path because only one object was hit by the shot ray. The path of the selected element also appears in the box. In this example, the object's path is `dom1/PENTA_6{TRI}`.



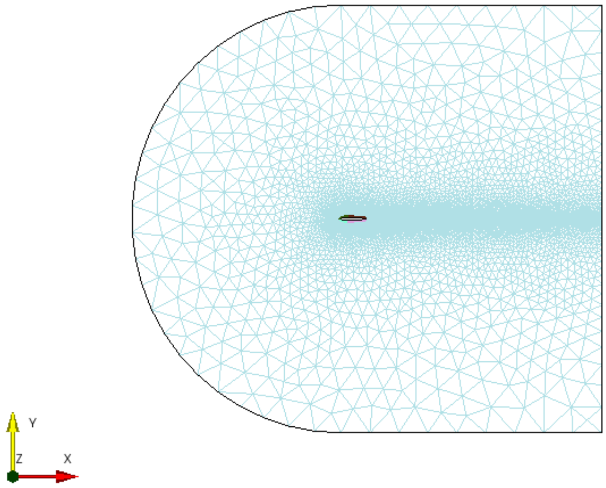
```
/base1/dom4 {imax}
/base1/dom4 {jmin}
/base1/dom4 {jmax}
/base1/dom4 {kmin}
/base1/dom4 {kmax}
/base1/dom4/ZoneBC/bcs_4_2
/base1dom1/PENTA_6 {QUAD}
dom1/PENTA_6 {TRI}
/base1dom1/QUAD_4 {QUAD}

dom1/PENTA_6 {TRI}
```

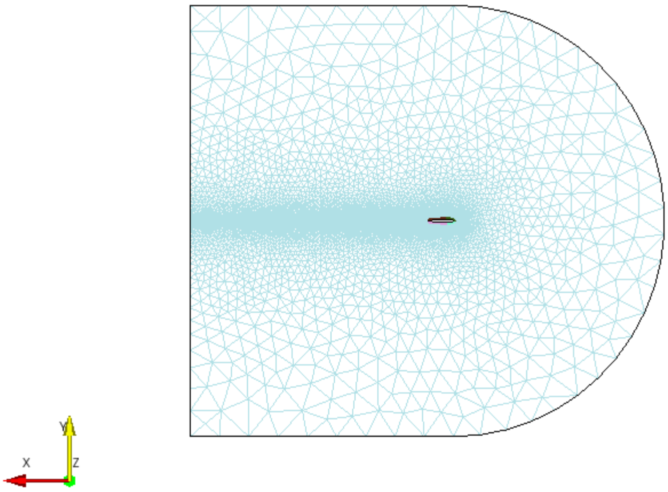
You can see that the paths of selected elements are marked by the icon ⏩ while the unselected elements are marked by the icon ◉.

Camera's position

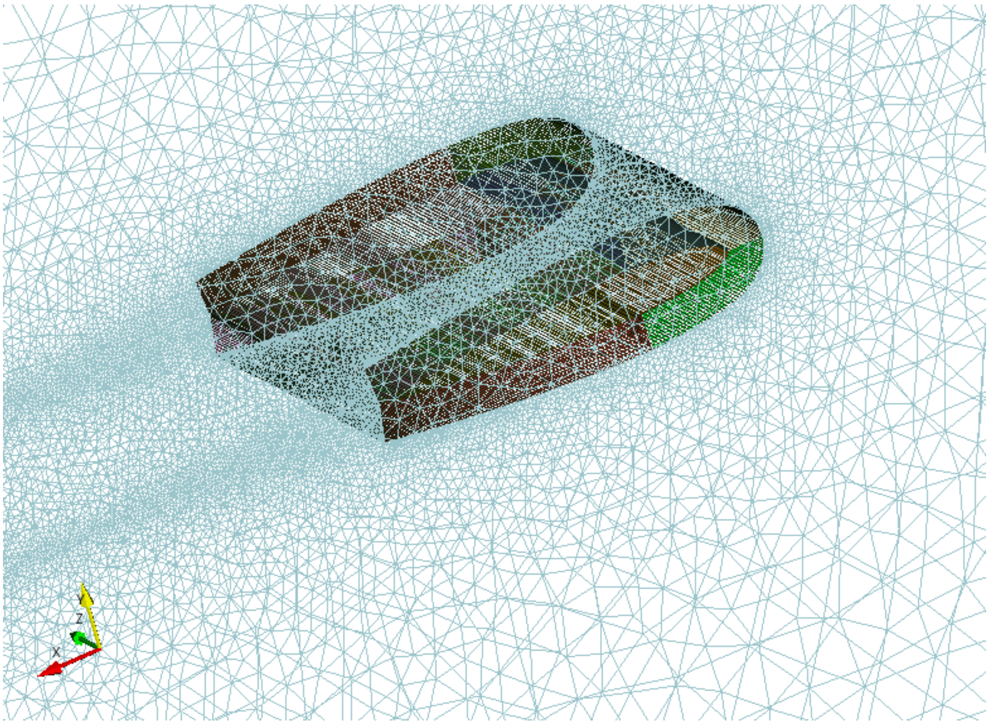
Icon	Action
X	Set the camera along the -X axis.
Y	Set the camera along the -Y axis.
Z	Set the camera along the -Z axis.
☑ Mirror + X	Set the camera along the +X, +Y or +Z axis if the mirror case is checked.
`Ctrl +` X	Rotate about the X,Y or Z direction.

If you want to set the viewing position to view the data along -Z axis, just click on Z. To display the view along the opposite direction, +Z axis, check ☑ Mirror and click on Z.

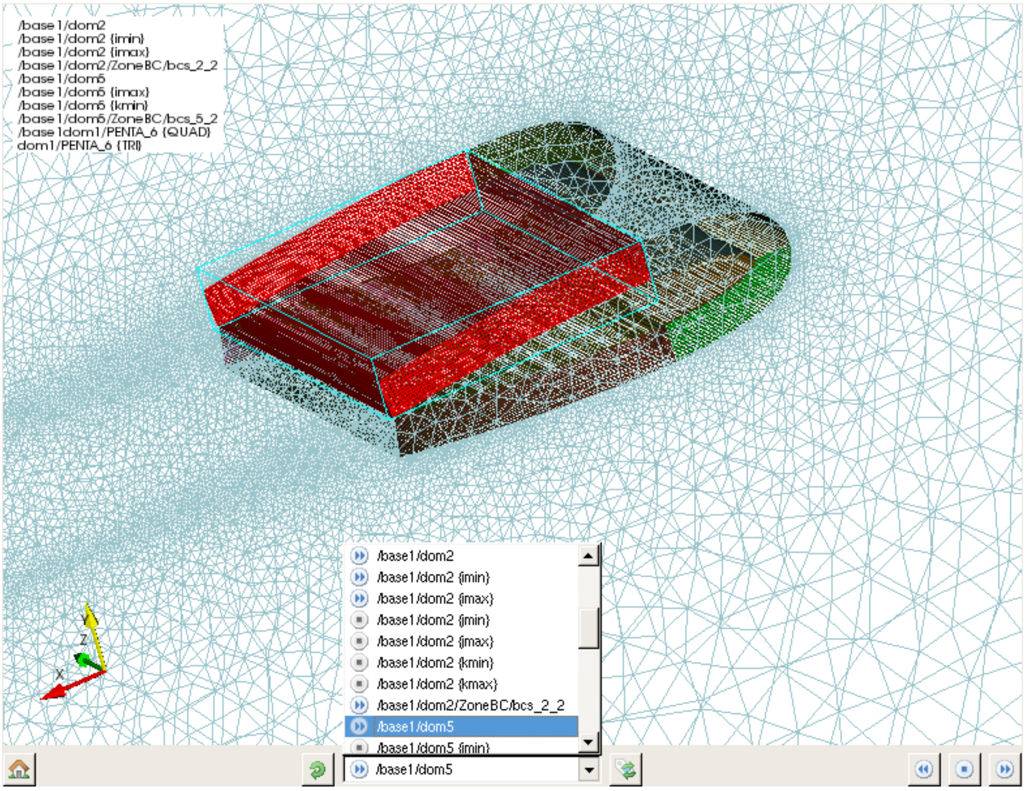




To display the other elements of the CGNS file, you have to handle the view with the mouse. By using the `Button 1` of the mouse to rotate the view, the `Button 2` to tranlate it and the `Button 3` with a motion of the mouse from bottom to up to scale up the elements, the view is adjusted like that:



and you perform a pick operation by pressing the p key:

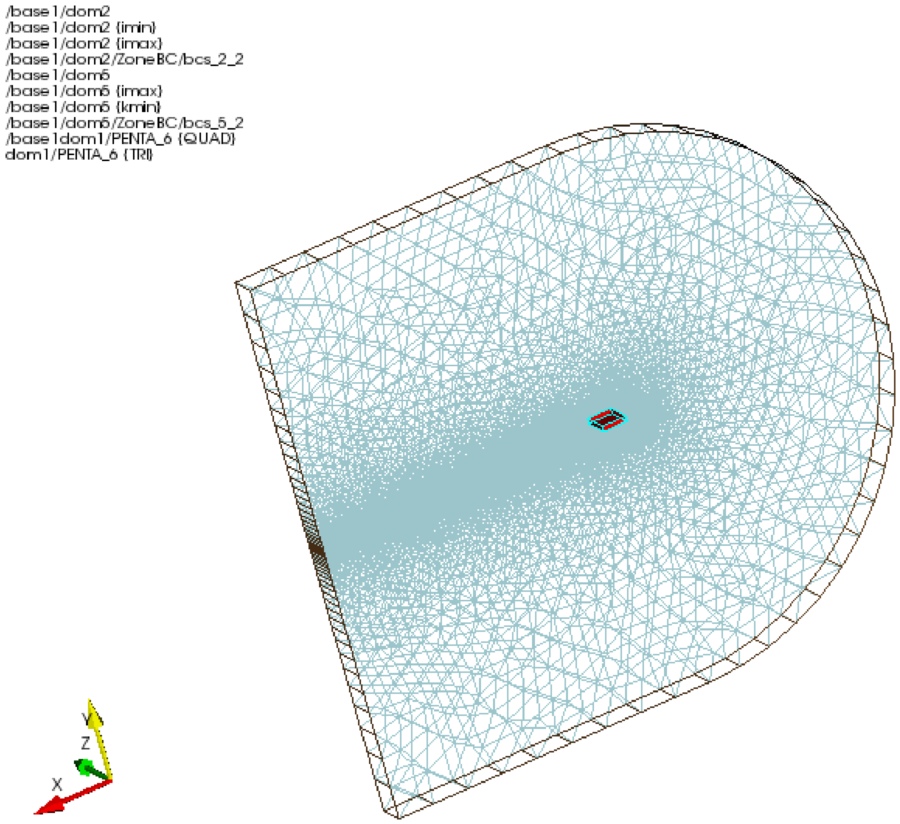


All the paths of the selected objects are displayed in the top left-hand corner of the window and they are marked with the ⏩ in the list below. As you can see, the current selected object is the same as previously, namely `dom1/PENTA_6{TRI}` because it's the closest object to the camera.

If a particular view interests you, you can save it and restore it later. When a desired view is achieved, type a view's name in the box Save view 1 ▾ located at the top of the window and press the `Enter` key or click the 📷 icon to save the view.

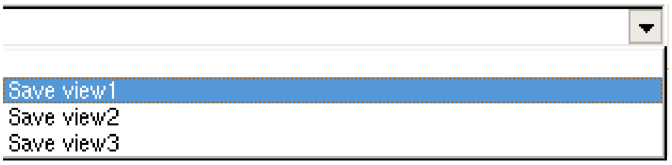
You can add a number of different views by repeating the previous operation.

Know, you want to have an overall view of the tree. To do that, press on the `r` key.



When you pressed the `r` key, the size of the objects changed to fit in the `VTK view`, the objects are centered while the camera keeps the current view direction.

To restore a saved view, choose the view's name in the list, like below:

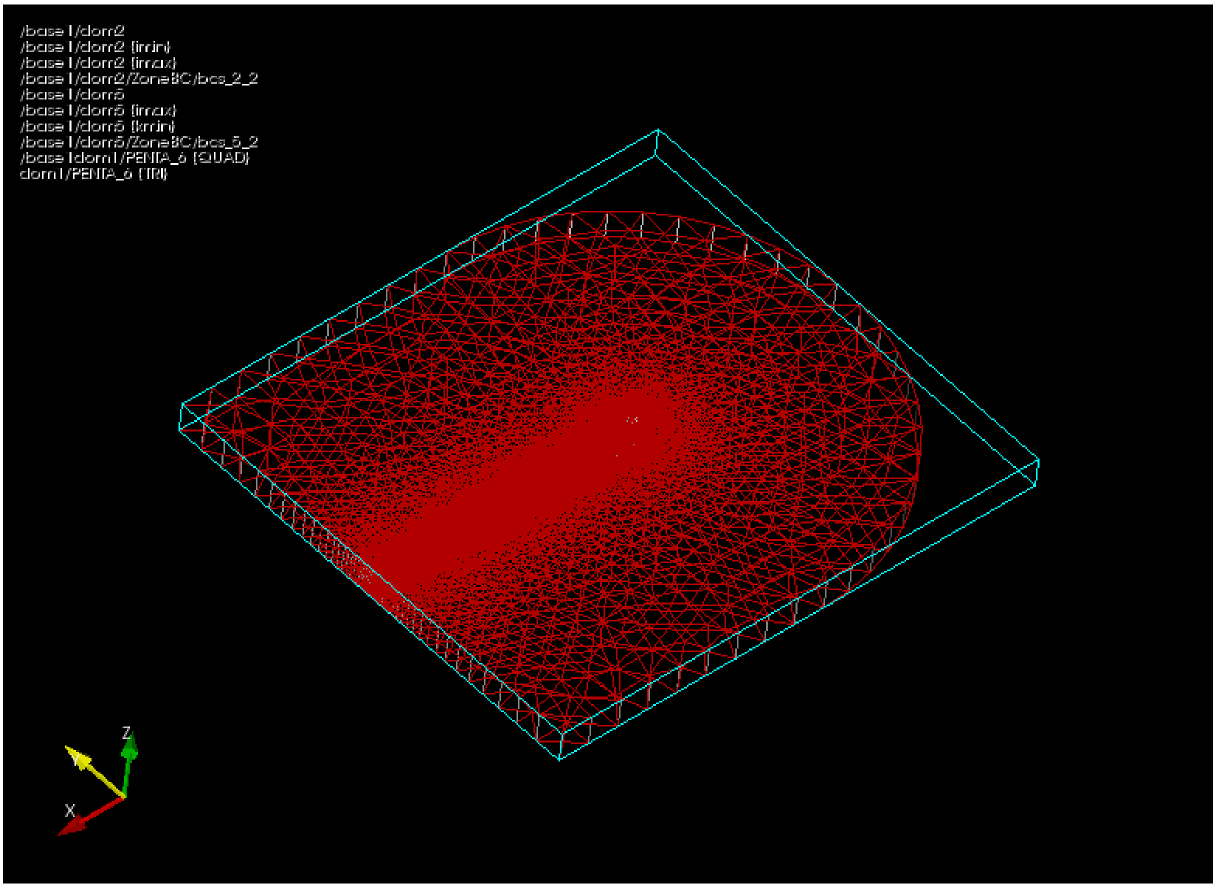


To delete an unwanted view, select the view's name and click the 📷 icon.

To save a view

Icon	Action
📷	Save the current view and add it to the view list.
📷	Write the view list into a file.
📷	Remove the current view from the view list.
Save view 1 ▾	Type here the view's name to save.

You can change the colours randomly by clicking the 🌈 icon and if you click the ◐ icon, the `VTK view` switch between a black background and a white background, like below:



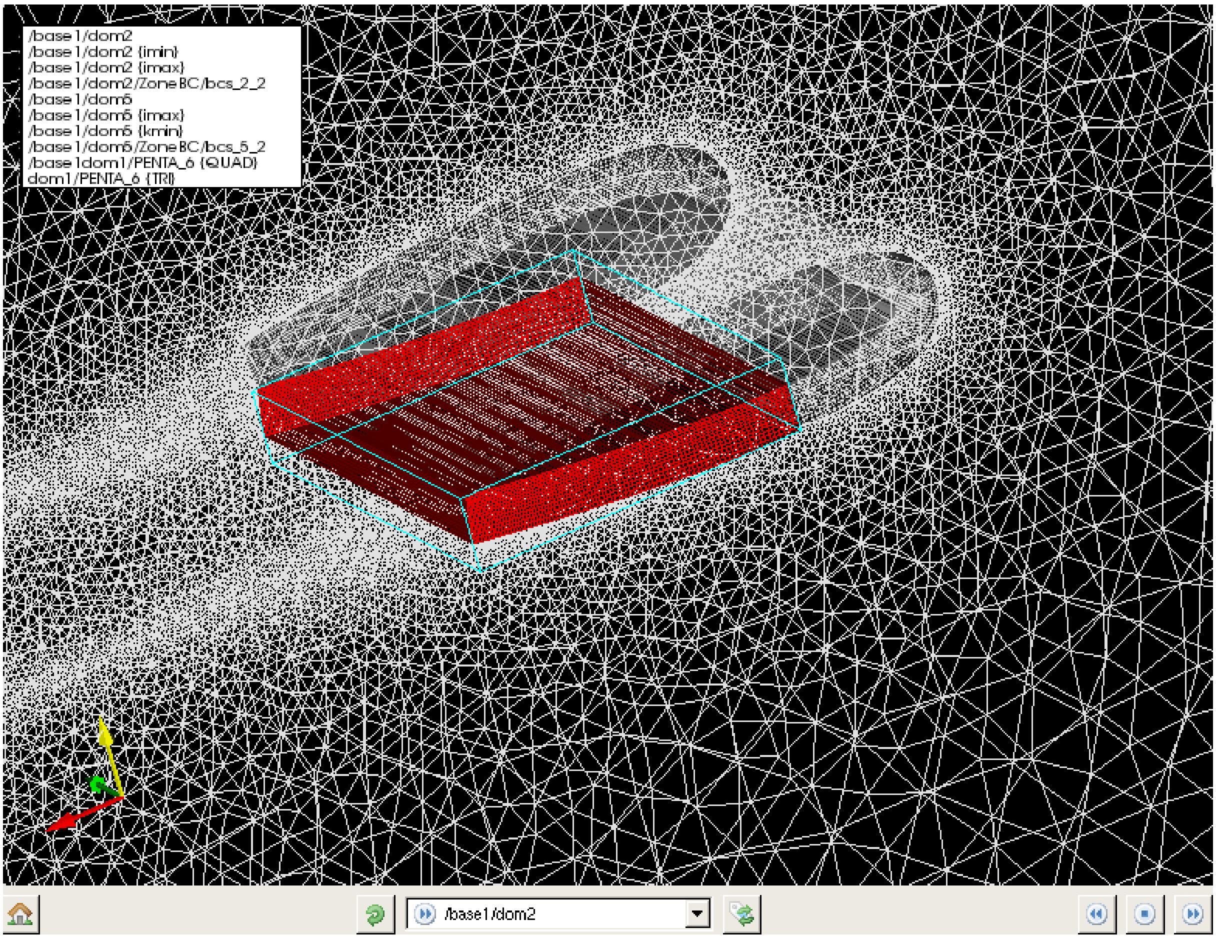
```
/base1/dom2
/base1/dom2 (imin)
/base1/dom2 (imax)
/base1/dom2/ZoneBC/bcs_2_2
/base1/dom5
/base1/dom5 (imax)
/base1/dom5 (kmin)
/base1/dom5/ZoneBC/bcs_5_2
/base1dom1/PENTA_6 (QUAD)
dom1/PENTA_6 (TRI)
```

When you changed the foreground of the `VTK view`, the current selected object remained selected because it still appears outlined by the blue wireframe bounding box and it is highlighted in red. It's also the case when you change the colors of objects.

To change colours

Icon	Action
🌈	Change the colours randomly.
◔	Switch between a black background and a white background.

You can also modify the current selected object by using the ⏩ icon and the ⏪ icon.



```
/base1/dom2
/base1/dom2 {imin}
/base1/dom2 {imax}
/base1/dom2/ZoneBC/bcs_2_2
/base1/dom5
/base1/dom5 {imax}
/base1/dom5 {kmin}
/base1/dom5/ZoneBC/bcs_5_2
/base1dom1/PENTA_6 {QUAD}
dom1/PENTA_6 {TRI}
```

The current selected object is here `/base1/dom2`.

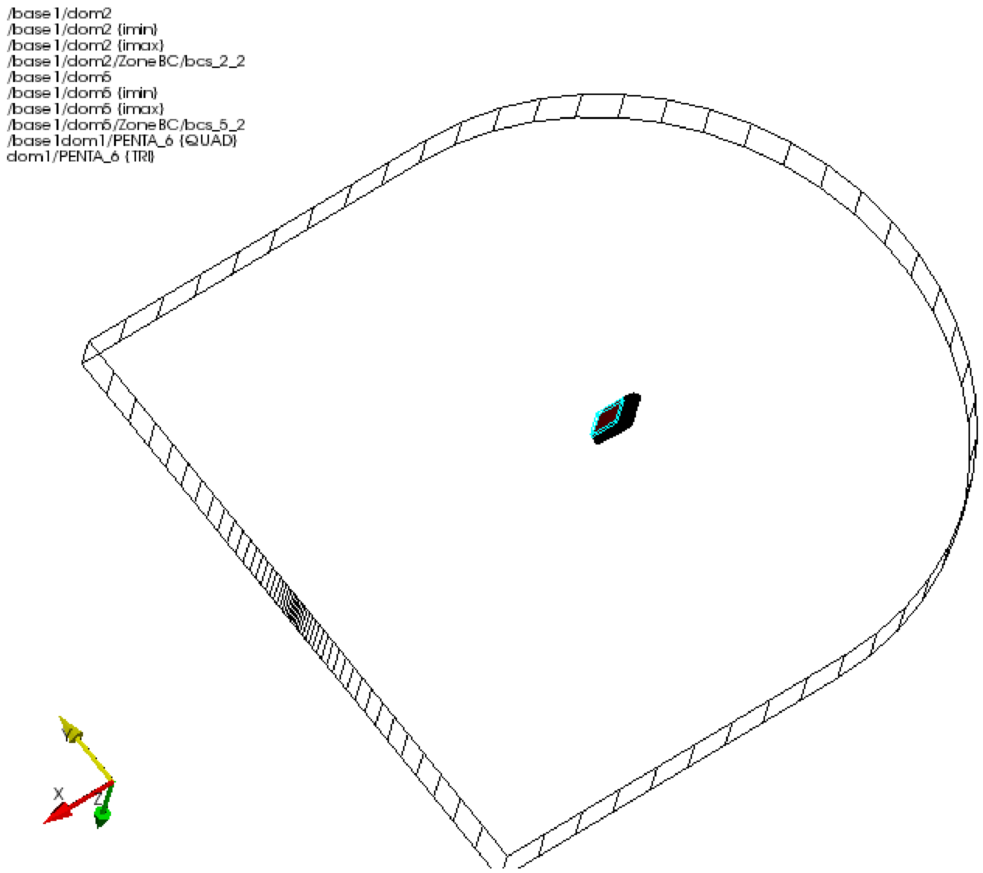
When you perform this operation, the next element ⏩ or the previous element ⏪ of the list which contains the objects picked becomes the current selected object. After the last element of the selection is reached, the first object of the list is again selected as current selected object.

To unselect all elements, click the ⊙ icon.

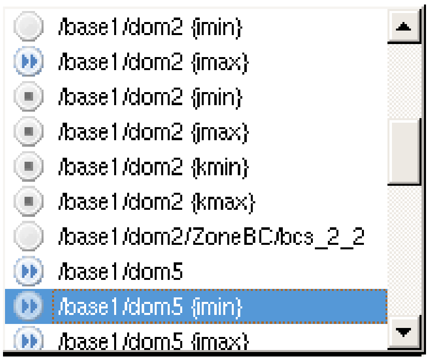
To modify the selected object

Icon	Action
⏩	Change the selected object by taking the following object in the selected objects list.
⏪	Change the selected object by taking the previous object in the selected objects list.
⊙	Set all elements as unselected objects and the hidden objects become visible.

To see better a part of the view, you can remove visible elements of the tree. Once you performed a pick operation to select objects, press the `d` key to hide the current selected element. After the objet is hidden, the next element located in the selected objects list becomes the current selected element. You can repeat the operation as long as list of the selected objects isn't empty.

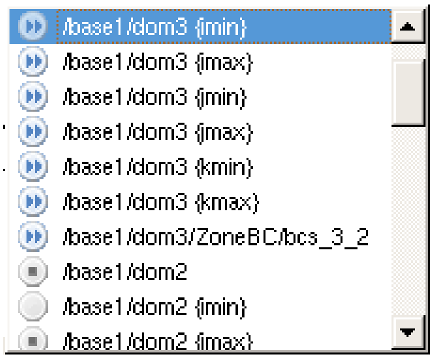


As you can observe it, the hidden objects are marked with the ○ icon.



○ /base1/dom2 {imin}
⏩ /base1/dom2 {imax}
■ /base1/dom2 {jmin}
■ /base1/dom2 {jmax}
■ /base1/dom2 {kmin}
■ /base1/dom2 {kmax}
○ /base1/dom2/ZoneBC/bcs_2_2
⏩ /base1/dom5
⏩ /base1/dom5 {imin}
⏩ /base1/dom5 {imax}

If you click the ⇄ button, you switch between the selected objects and the unselected objects. The hidden objects remain unchanged.

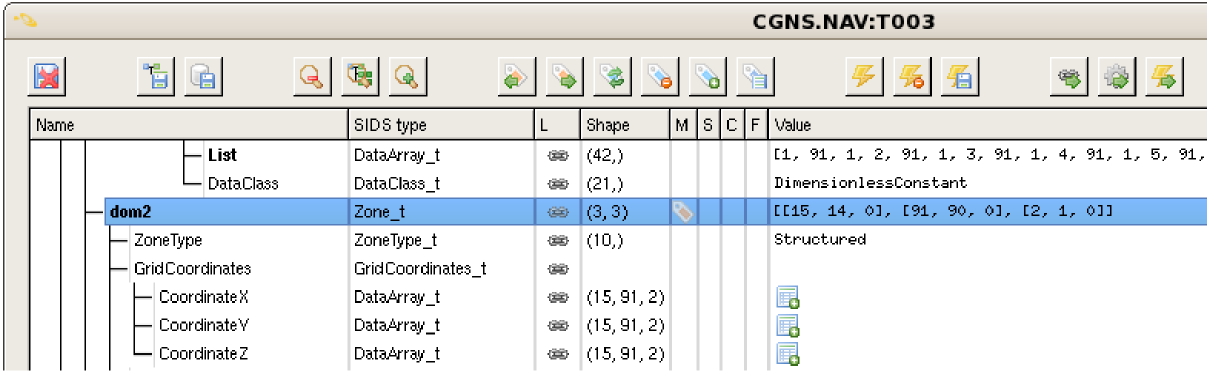


⏩ /base1/dom3 {imin}
⏩ /base1/dom3 {imax}
⏩ /base1/dom3 {jmin}
⏩ /base1/dom3 {jmax}
⏩ /base1/dom3 {kmin}
⏩ /base1/dom3 {kmax}
⏩ /base1/dom3/ZoneBC/bcs_3_2
■ /base1/dom2
○ /base1/dom2 {imin}
■ /base1/dom2 {imax}

When you click the icon, all elements become unselected and the objects which are hidden become again visible. Consequently, all elements of the list are marked with icon.

To display only a part of the CGNS tree

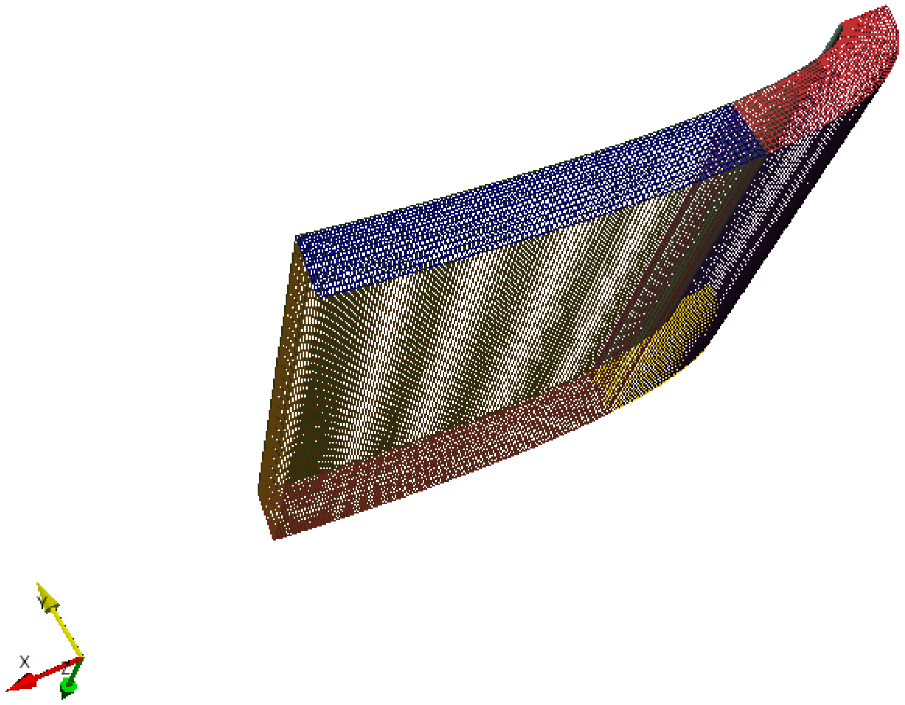
The CGNS/tree of our example displayed in the `VTK view` is composed of several zones, namely five zones. You can choose to visualize only a part of the tree by marcking the nodes which you want to display. Click on its name and press on the `Space` key to select a node.



CGNS.NAV:T003

Name	SIDS type	L	Shape	M	S	C	F	Value
List	DataArray_t		(42,)					[1, 91, 1, 2, 91, 1, 3, 91, 1, 4, 91, 1, 5, 91,
DataClass	DataClass_t		(21,)					DimensionlessConstant
dom2	Zone_t		(3, 3)	🏷				[[15, 14, 0], [91, 90, 0], [2, 1, 0]]
ZoneType	ZoneType_t		(10,)					Structured
GridCoordinates	GridCoordinates_t							
CoordinateX	DataArray_t		(15, 91, 2)					
CoordinateY	DataArray_t		(15, 91, 2)					
CoordinateZ	DataArray_t		(15, 91, 2)					

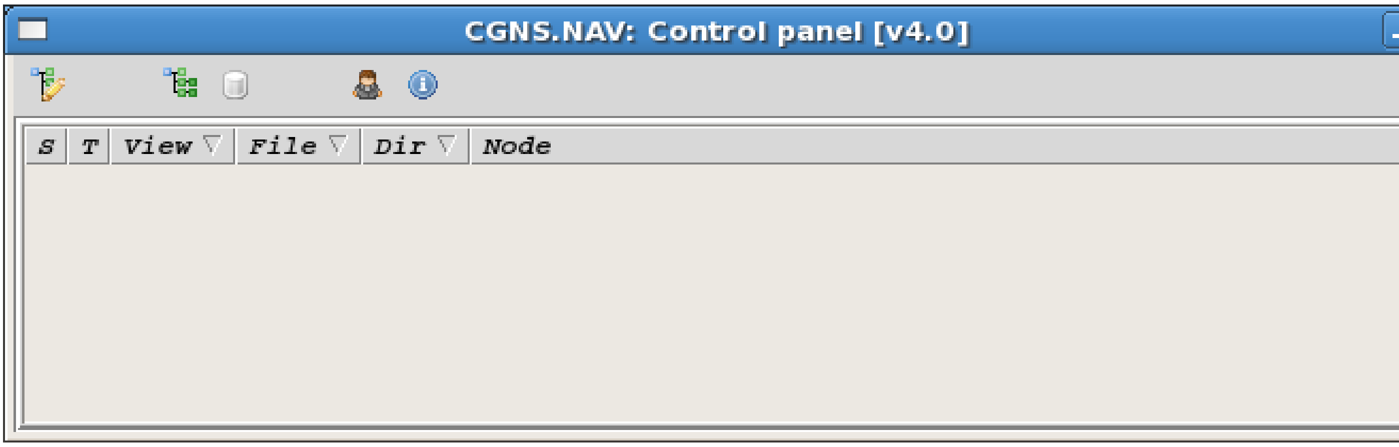
As you can observe it, the selected node is marked with the flag. Then, all you have to do is click the icon and the following `VTK view` appears:



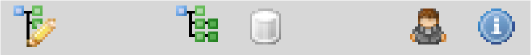
Only the `dom2` zone is dispayed in the `VTK view`. You can add the other zones to be visualized by performing the same operation. The zones which contain at least one marked node will be displayed.

3.1 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.

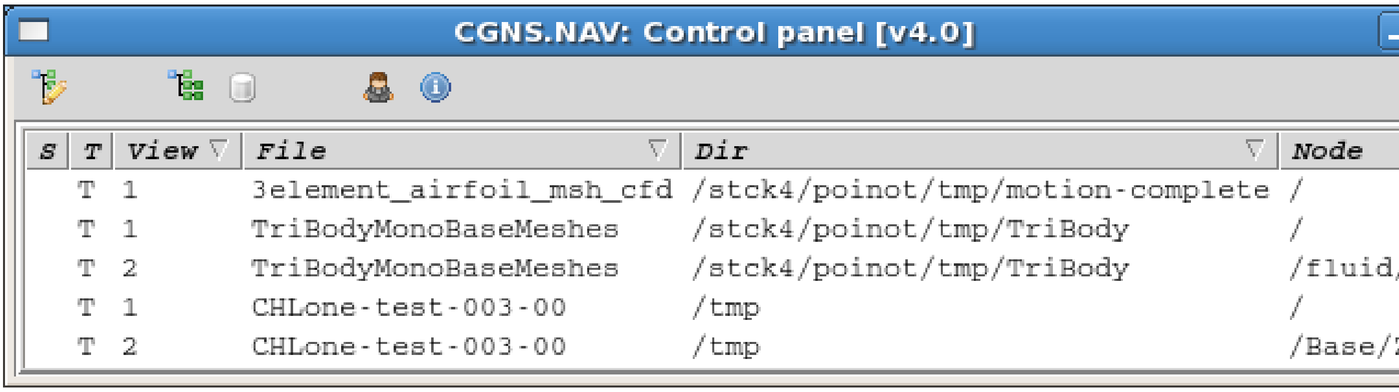


If you select the line of an existing view in the *Control view* and press *Return*, this view will be raised on your screen. You can find back the *Control view* from any other view just pressing the *BackSpace* key (See *Key/Mouse Bindings*).



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.

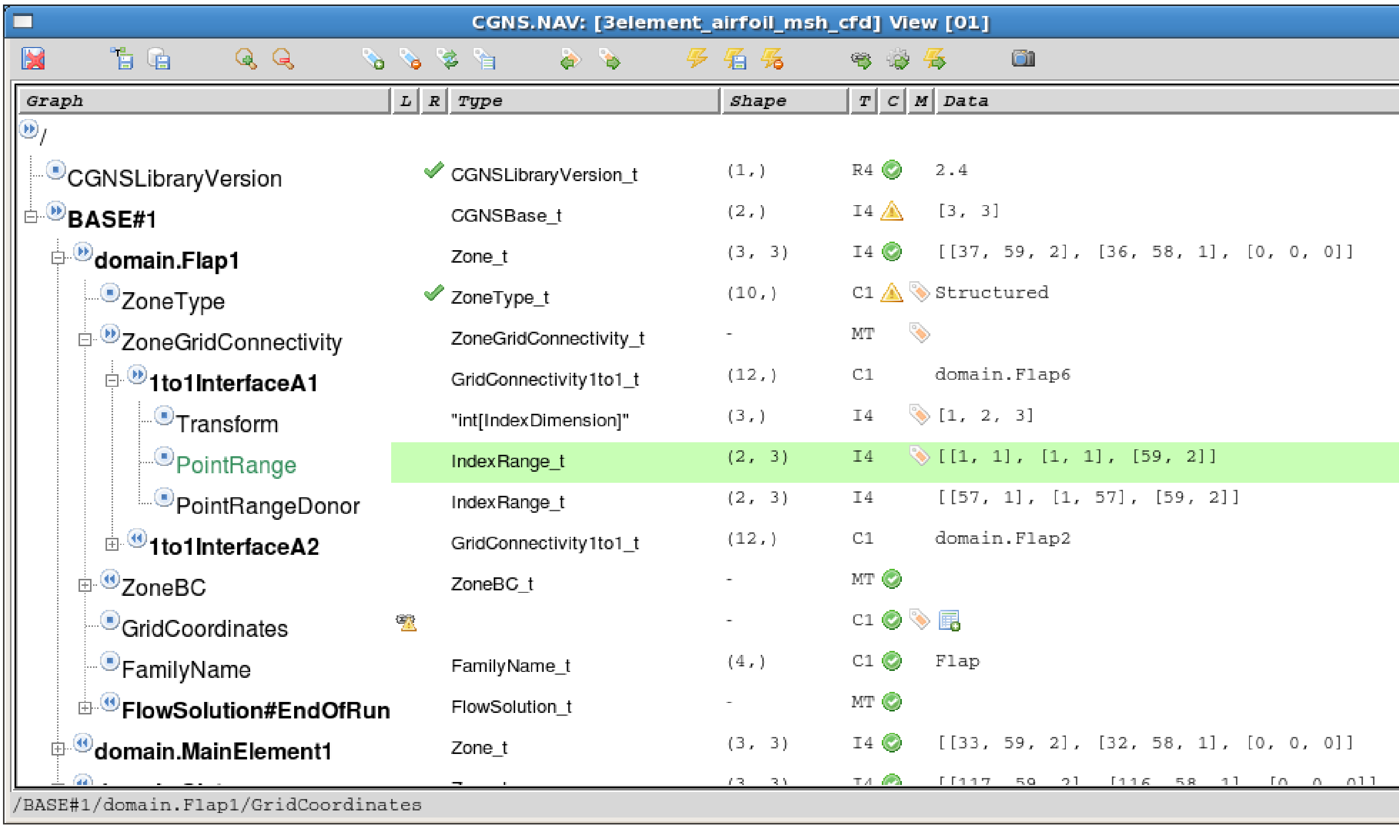


S	T	View	File	Dir	Node
	T	1	3element_airfoil_msh_cfd	/stck4/poinot/tmp/motion-complete	/
	T	1	TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/
	T	2	TriBodyMonoBaseMeshes	/stck4/poinot/tmp/TriBody	/fluid
	T	1	CHLone-test-003-00	/tmp	/
	T	2	CHLone-test-003-00	/tmp	/Base/

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 youhave 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.

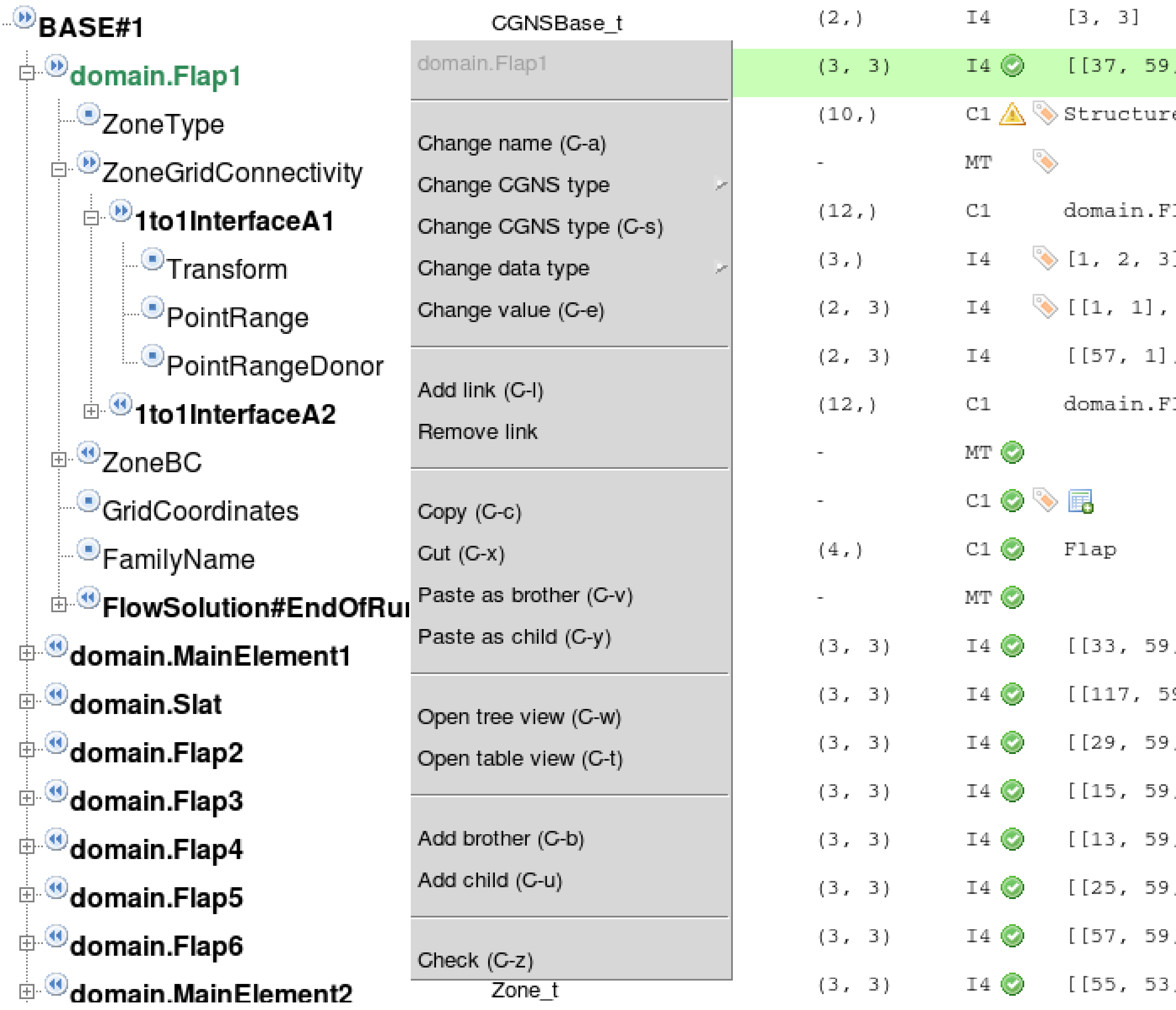


CGNS.NAV: [3element_airfoil_msh_cfd] View [01]

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.Flap1			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.Flap6
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.Flap2
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]]

/BASE#1/domain.Flap1/GridCoordinates

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 simpple 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.

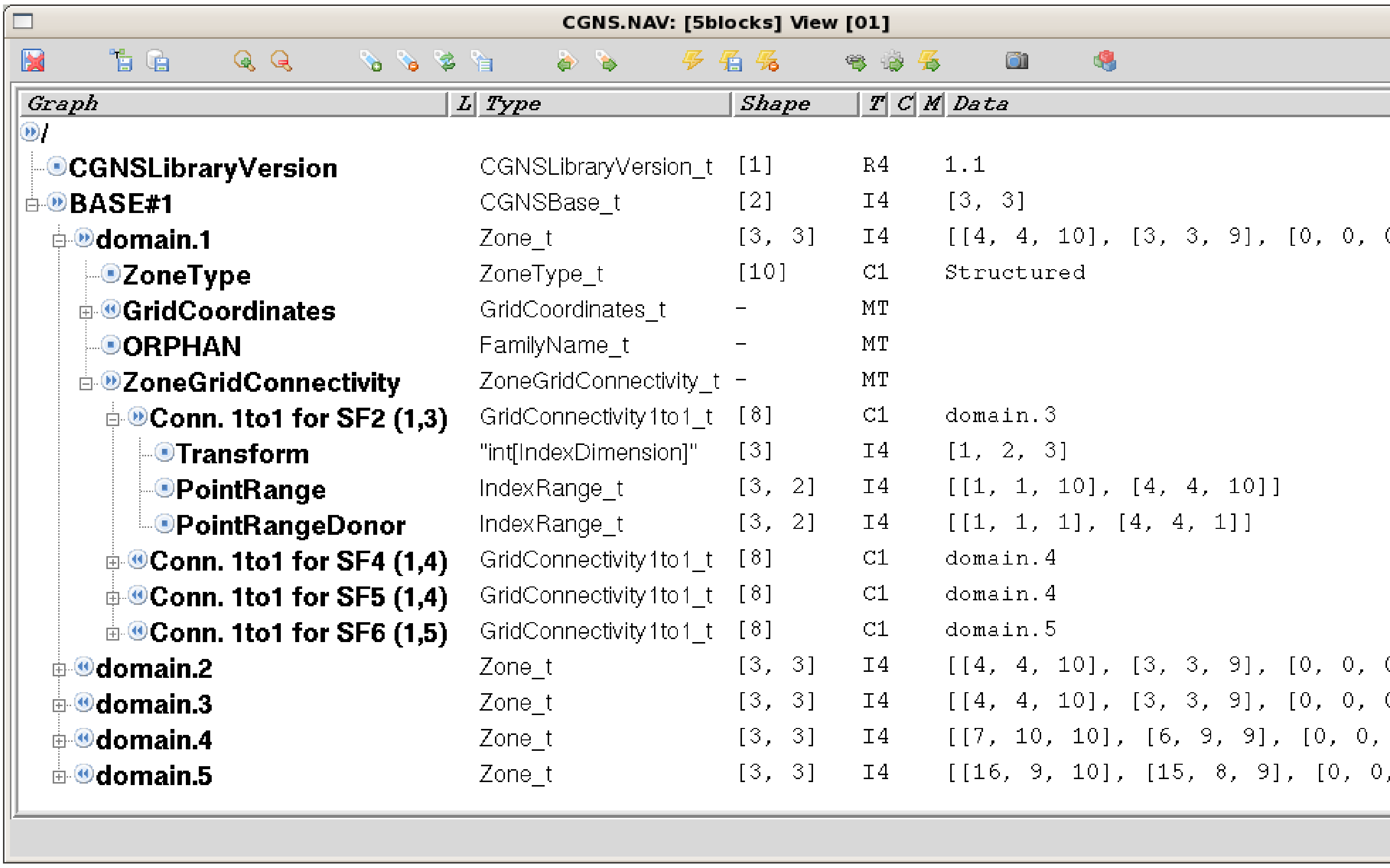


BASE#1	CGNSBase_t	(2,)	I4	[3, 3]
domain.Flap1		(3, 3)	I4 ✓	[[37, 59
ZoneType		(10,)	C1 ⚠	Structure
ZoneGridConnectivity		-	MT	
1to1InterfaceA1		(12,)	C1	domain.F
Transform		(3,)	I4	[1, 2, 3
PointRange		(2, 3)	I4	[[1, 1],
PointRangeDonor		(2, 3)	I4	[[57, 1]
1to1InterfaceA2		(12,)	C1	domain.F
ZoneBC		-	MT ✓	
GridCoordinates		-	C1 ✓	
FamilyName		(4,)	C1 ✓	Flap
FlowSolution#EndOfRu		-	MT ✓	
domain.MainElement1		(3, 3)	I4 ✓	[[33, 59
domain.Slat		(3, 3)	I4 ✓	[[117, 59
domain.Flap2		(3, 3)	I4 ✓	[[29, 59
domain.Flap3		(3, 3)	I4 ✓	[[15, 59
domain.Flap4		(3, 3)	I4 ✓	[[13, 59
domain.Flap5		(3, 3)	I4 ✓	[[25, 59
domain.Flap6		(3, 3)	I4 ✓	[[57, 59
domain.MainElement2	Zone_t	(3, 3)	I4 ✓	[[55, 53

3.2 Tree View

The first window `CGNS.NAV` opens when you read a *CGNS* file is the *Tree view*. This is you 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.

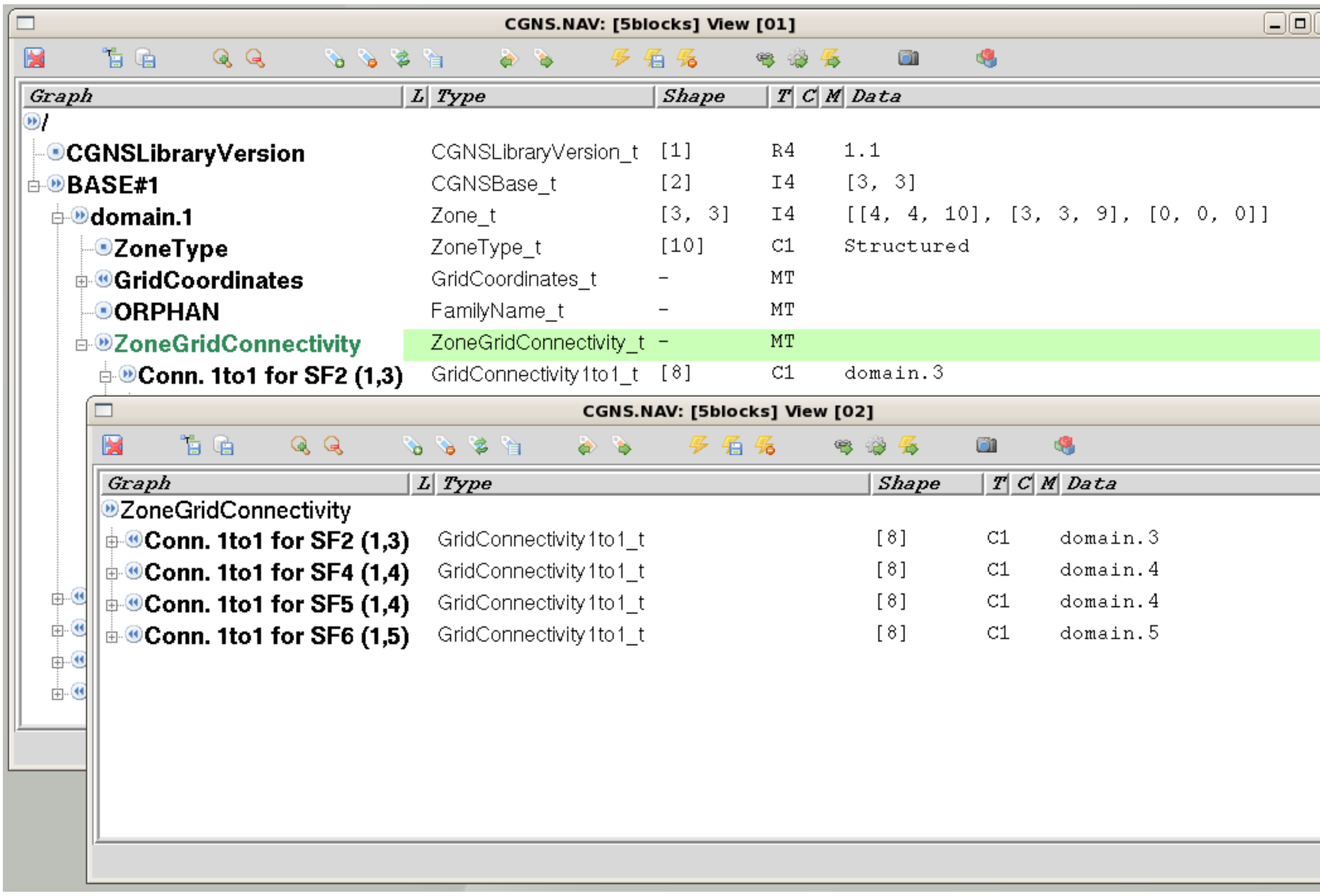


CGNS.NAV: [5blocks] View [01]

Graph	L	Type	Shape	T	C	M	Data
/							
CGNSLibraryVersion		CGNSLibraryVersion_t	[1]	R4			1.1
BASE#1		CGNSBase_t	[2]	I4			[3, 3]
domain.1		Zone_t	[3, 3]	I4			[[4, 4, 10], [3, 3, 9], [0, 0, (
ZoneType		ZoneType_t	[10]	C1			Structured
GridCoordinates		GridCoordinates_t	-	MT			
ORPHAN		FamilyName_t	-	MT			
ZoneGridConnectivity		ZoneGridConnectivity_t	-	MT			
Conn. 1to1 for SF2 (1,3)		GridConnectivity1to1_t	[8]	C1			domain.3
Transform		"int[IndexDimension]"	[3]	I4			[1, 2, 3]
PointRange		IndexRange_t	[3, 2]	I4			[[1, 1, 10], [4, 4, 10]]
PointRangeDonor		IndexRange_t	[3, 2]	I4			[[1, 1, 1], [4, 4, 1]]
Conn. 1to1 for SF4 (1,4)		GridConnectivity1to1_t	[8]	C1			domain.4
Conn. 1to1 for SF5 (1,4)		GridConnectivity1to1_t	[8]	C1			domain.4
Conn. 1to1 for SF6 (1,5)		GridConnectivity1to1_t	[8]	C1			domain.5
domain.2		Zone_t	[3, 3]	I4			[[4, 4, 10], [3, 3, 9], [0, 0, (
domain.3		Zone_t	[3, 3]	I4			[[4, 4, 10], [3, 3, 9], [0, 0, (
domain.4		Zone_t	[3, 3]	I4			[[7, 10, 10], [6, 9, 9], [0, 0,
domain.5		Zone_t	[3, 3]	I4			[[16, 9, 10], [15, 8, 9], [0, 0,

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 [1]).

[1] All key bindings can be customized, the bindings we give here are the *default bindings*.



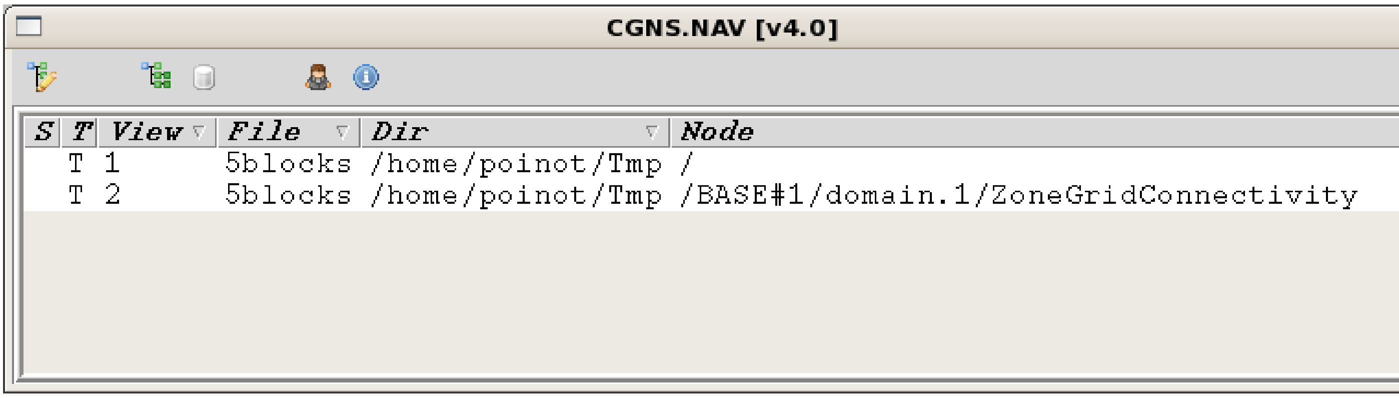
CGNS.NAV: [5blocks] View [01]

Graph	L	Type	Shape	T	C	M	Data
/							
CGNSLibraryVersion		CGNSLibraryVersion_t	[1]	R4			1.1
BASE#1		CGNSBase_t	[2]	I4			[3, 3]
domain.1		Zone_t	[3, 3]	I4			[[4, 4, 10], [3, 3, 9], [0, 0, 0]]
ZoneType		ZoneType_t	[10]	C1			Structured
GridCoordinates		GridCoordinates_t	-	MT			
ORPHAN		FamilyName_t	-	MT			
ZoneGridConnectivity		ZoneGridConnectivity_t	-	MT			
Conn. 1to1 for SF2 (1,3)		GridConnectivity1to1_t	[8]	C1			domain.3

CGNS.NAV: [5blocks] View [02]

Graph	L	Type	Shape	T	C	M	Data
ZoneGridConnectivity							
Conn. 1to1 for SF2 (1,3)		GridConnectivity1to1_t	[8]	C1			domain.3
Conn. 1to1 for SF4 (1,4)		GridConnectivity1to1_t	[8]	C1			domain.4
Conn. 1to1 for SF5 (1,4)		GridConnectivity1to1_t	[8]	C1			domain.4
Conn. 1to1 for SF6 (1,5)		GridConnectivity1to1_t	[8]	C1			domain.5

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



CGNS.NAV [v4.0]

S	T	View	File	Dir	Node
	T	1	5blocks	/home/poinot/Tmp	/
	T	2	5blocks	/home/poinot/Tmp	/BASE#1/domain.1/ZoneGridConnectivity

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.

3.2.1 The icon menu

3.2.2 Selection and flags

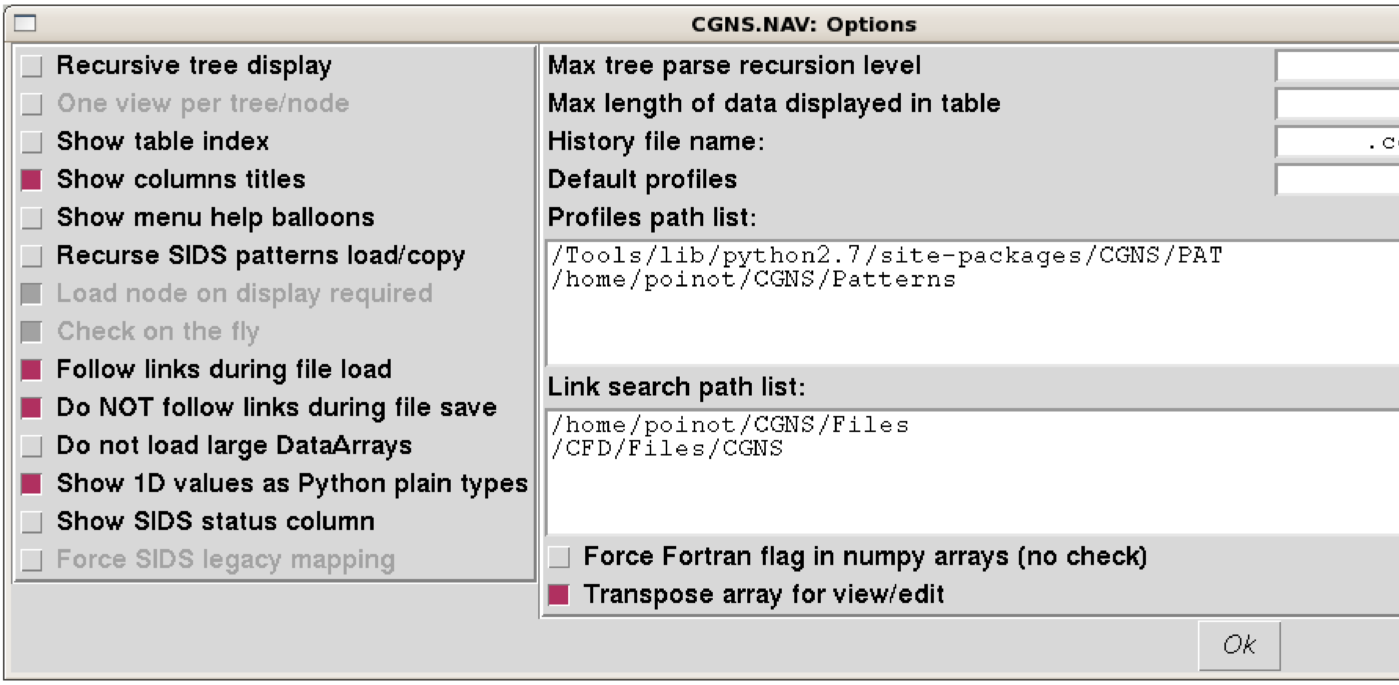
3.2.3 Copy/Paste

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 (usinf <*Control-V*> (see [2]).

3.3 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.



CGNS.NAV: Options

□ Recursive tree display
□ One view per tree/node
□ Show table index
☑ Show columns titles
□ Show menu help balloons
□ Recurse SIDS patterns load/copy
☑ Load node on display required
☑ Check on the fly
☑ Follow links during file load
☑ Do NOT follow links during file save
□ Do not load large DataArrays
☑ Show 1D values as Python plain types
□ Show SIDS status column
□ Force SIDS legacy mapping

Max tree parse recursion level
Max length of data displayed in table
History file name: .c
Default profiles
Profiles path list:

```
/Tools/lib/python2.7/site-packages/CGNS/PAT
/home/poinot/CGNS/Patterns
```

Link search path list:

```
/home/poinot/CGNS/Files
/CFD/Files/CGNS
```

□ Force Fortran flag in numpy arrays (no check)
☑ Transpose array for view/edit

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.

3.3.1 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.

[2] 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

- ***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 yelllow 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*

3.3.2 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
Con-trolView	<Return> <Delete>	On a view line, raises this view window On a view line, closes the window and its children windows
Tree-View	<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 *Tree window* for selected node Opens a new *Table window* 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
Table-View	<Backspace>	Raises the ControlView window

3.4 Pattern View

3.5 VTK View

3.5.1 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.

3.5.2 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

3.6 Query View

3.7 Link View

3.8 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.

CHAPTER

FOUR

GLOSSARY

cgns.org The official CGNS web site, by extension any document on this web site has an *official* taste...

CGNS The specific purpose of the CFD General Notation System (CGNS) project is to provide a standard for recording and recovering computer data associated with the numerical solution of the equations of fluid dynamics. See also the *How to?*.

CGNS/SIDS The Standard Interface Data Structure is the specification of the data model. This public document describes the syntax and the semantics of all tree-structured data required or proposed for a CFD simulation.

CGNS/MLL The Mid-Level Library is an example implementation of *CGNS/SIDS* on top of *CGNS/ADF* and *CGNS/HDF5* mappings. This library has a C and a Fortran API.

CGNS/ADF The Advanced Data Format *CGNS/SIDS* implementation. A binary storage format and its companion library, developped by *Boeing*.

CGNS/HDF5 The Hierarchical Data Format *CGNS/SIDS* implementation. A binary storage format and its companion library (see below).

CGNS/Python The Python programming language *CGNS/SIDS* implementation.

CHLone A *CGNS/HDF5* compliant implementation. The CHLone library is available on SourceForge.

HDF5 A powerful storage system for large data. The HDF5 library should be seen as a middleware system with a lot of powerful features related to efficient, portable and trustable storage mean.

python An object oriented interpreted programming language.

cython A compiler tool that translate Python/Numpy into C code for performance purpose.

numpy The numerical library for Python. *Numpy* is used to store the data in Python arrays which have a direct memory mapping to actual C or Fortran memory.

VTK A visualization toolkit used to display 3D objects ni *CGNS.NAV*.

PySide The Python interface for the Qt toolkit. PySide

Qt A powerful graphical toolkit available under GPL v3, LGPL v2 and a commercial license. The current use of Qt is under LGPL v2 in pyCGNS.

4.1 NAV Index

- *genindex*