



BETA

SIMULATION SOLUTIONS



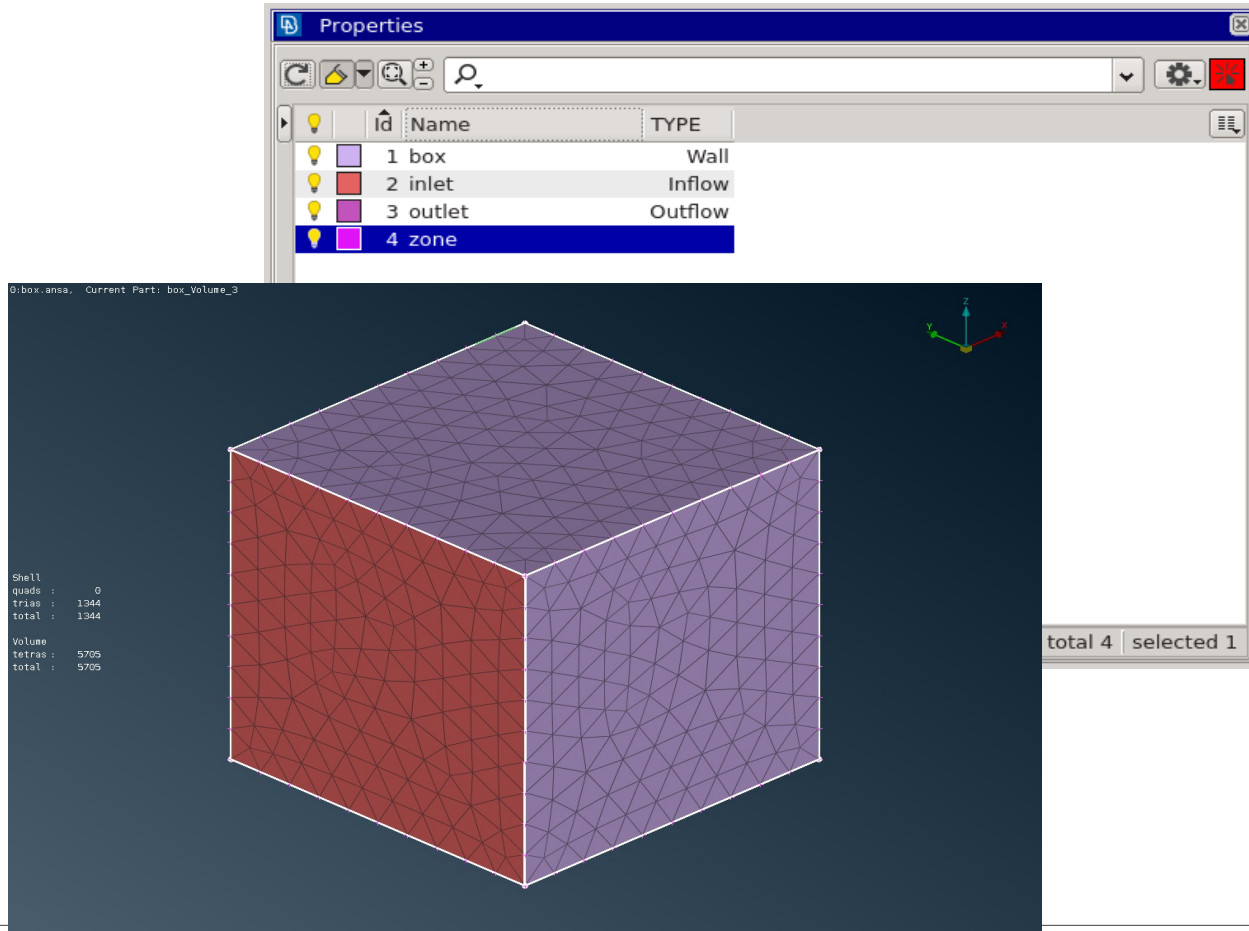
**Groundbreakin
Simulation Solutions**

physics on screen

CGNS format:
is portability really there?

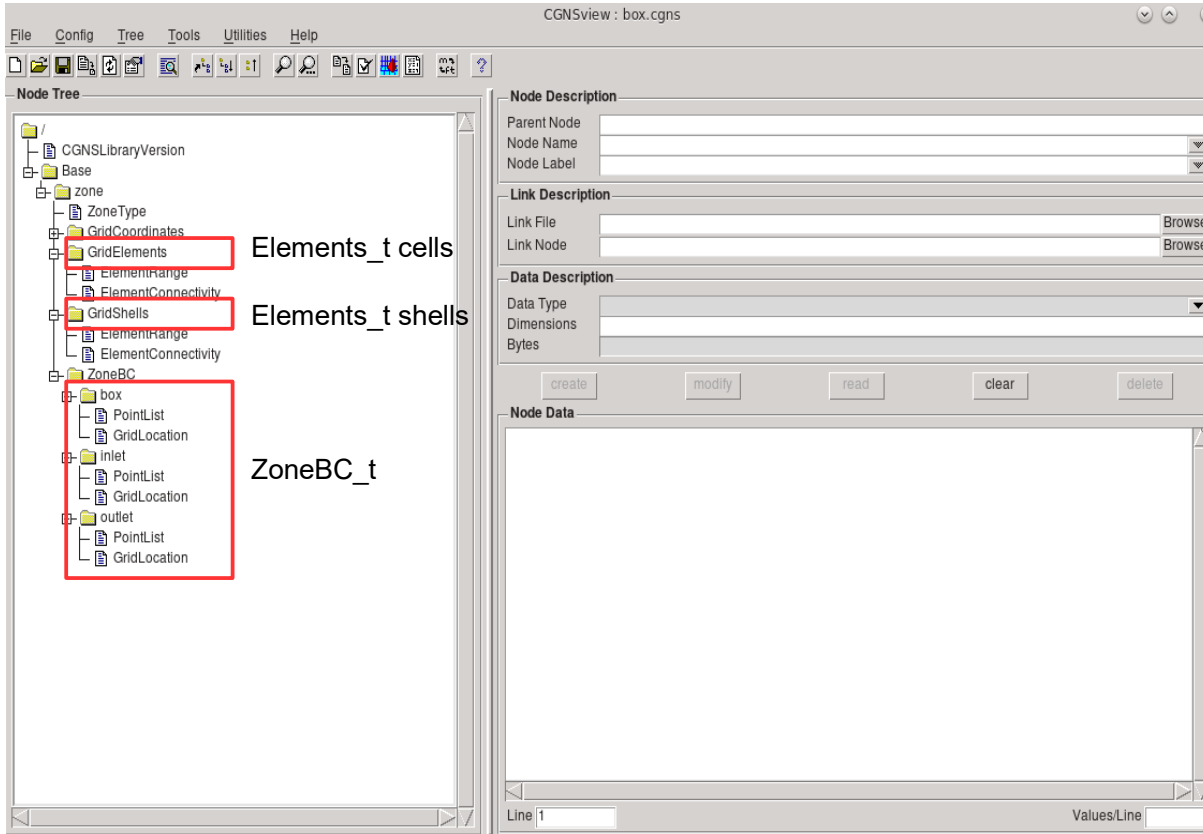
- Many file setups using different CGNS node combinations are valid, e.g.
 - ✓ One or many Elements_t nodes (mixed, separated, NGON)
 - ✓ PointList or PointRange
 - ✓ Usage of Family_t node or not
- ZoneBC_t node must always be there.
- Each software develops one or more combinations.
- Not all CFD software is synchronized with the new CGNS features: some software is not following the new devs, some other is moving slow towards them and another is up-to-date.

CGNS format:
Current state

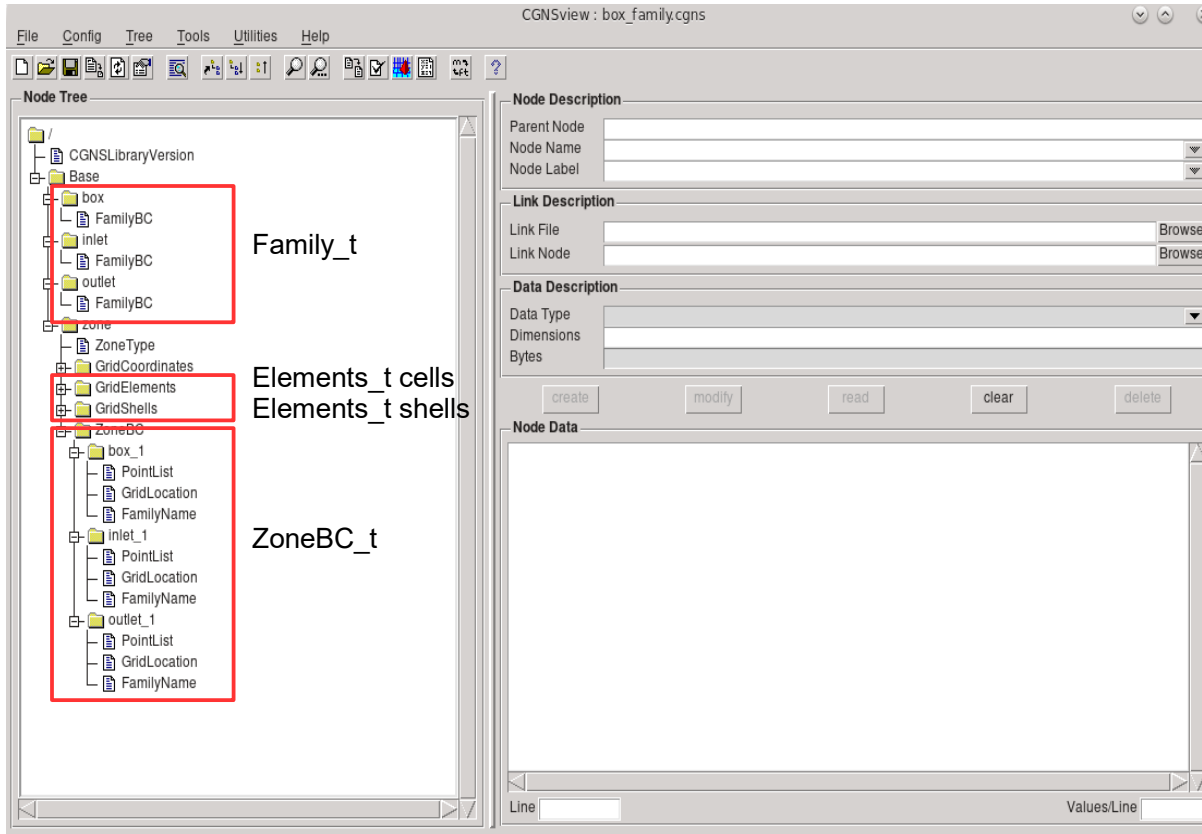


CGNS format: the problem

What happens when
a file is shared among
different software?



Scenario #1



Scenario #2

Scenario #3

The screenshot shows the CGNSview application window titled "CGNSview : box_bccorr.cgns". The interface is divided into several sections:

- Node Tree:** A hierarchical tree view on the left. A red box highlights the following structure:
 - zone
 - ZoneType
 - GridCoordinates
 - GridElements (highlighted)
 - box
 - ElementRange
 - ElementConnectivity
 - inlet
 - ElementRange
 - ElementConnectivity
 - outlet
 - ElementRange
 - ElementConnectivity
 - ZoneBC (highlighted)
 - box
 - PointList
 - GridLocation
 - inlet
 - PointList
 - GridLocation
 - outlet
 - PointList
 - GridLocation

Annotations on the right side of the Node Tree:

- "Elements_t cells" points to the **GridElements** node.
- "Elements_t shells named after the bcs" points to the **inlet** and **outlet** sub-nodes under **ZoneBC**.
- "ZoneBC_t" points to the **ZoneBC** node.

- Node Description:** Fields for Parent Node, Node Name, and Node Label.
- Link Description:** Fields for Link File (with a "Browse" button) and Link Node (with a "Browse" button).
- Data Description:** Fields for Data Type, Dimensions, and Bytes.
- Actions:** Buttons for create, modify, read, clear, and delete.
- Node Data:** A large empty text area for displaying node data.
- Line:** A field for the current line number.
- Values/Line:** A field for the number of values per line.

Most of our users are taking this approach

CGNS format: the concerns

- Lets consider the cgns file written from scenario #2.
What happens in the vendors'/custom parsers' side?
This *valid* cgns file fails to be read if :
 - ✗ families are not supported yet
 - ✗ one section per bc is required (why? - Is it PointRange that facilitates this? Are the sample files that guide the users there?)
 - ▶ Consider billion meshes cases where this will not be feasible at all due to memory issues
 - ✗ Zones_BC node is skipped, due to section naming conventions.

CGNS format: the solution

- We need to consider syncing our software to a minimum compliance level and have a uniform way of communicating it.
- We need to provide tools that allow API users to transform their valid CGNS file representation to a different one.

Thank you for your attention



Stay connected

