# CGNS Proposal Extension #0048: Monitoring definition and storage

Main authors: Mickael Philit (mickael.philit@safrangroup.com)

## Table of contents

1. Motivation ....1
2. Proposal to add a concept of Monitor/Probes ....1
3. Conflict and compatibility concern ....3
4. Conclusion ....3
5. Document modification list ....3
A. Appendix - Extension to the CGNS/SIDS ....4
B. Appendix - Extension to the CGNS/MLL ....4
C. Appendix – Document modification list ....4

## 1. Motivation

Most CFD solvers have the capability to extract physical information at specific location during the computation. This allows monitoring the convergence history or the unsteady fluctuations depending on the usecase. Existing structures of CGNS are not meant to handle such data and thus each solver use its own UserDefinedData_t node to do the storage.

Nowadays, it is quite easy to get access through CGNS standard to 3D states of the solution but there is lack concerning monitoring numerical probes. This proposal is here to fill the gap with a general and dedicated CGNS structure.

## 2. Proposal to add a concept of Monitor/Probes

### 2.1. Monitoring container node

The type **Monitor_t** is introduced in order to store the probes description and should have a basic property:

- **NumberOfSamples**: The total number of iteration or timesteps that can be stored in the children probes.

```
Monitor_t :=
{

   int NumberOfSamples;                                                      (r/d)

   List( Probe_t Probe1 ... ProbeN ) ;                                       (o)

   List( Descriptor_t Descriptor1 ... DescriptorN ) ;                        (o)
   List( UserDefinedData_t UserDefinedData1 ... UserDefinedDataN ) ;         (o)
} ;
```

The **Monitor_t** node is located under a base. Its children are Probes nodes.

The **NumberOfSamples** default value will be taken from BaseIterativeData_t as NumberOfSteps if the BaseIterativeData_t node exists else it should be defined.

## 1.1. Probes definition

To define a numerical probe, the following **Probe_t** structure is proposed:

```
Probe_t<NumberOfSamples> :=
{

  int Sampling                                                     (o/d)
                                                                   (r)
  List( DataArray_t<DataType, 1, Samples>
        DataArray1 ... DataArrayN ) ;                              (o/d)

  ProbeLocation_t Location                                         (o)
  FamilyName_t FamilyName                                          (o)
  List( Descriptor_t Descriptor1 ... DescriptorN ) ;               (o)
  List( UserDefinedData_t UserDefinedData1 ... UserDefinedDataN ) ;  (o)
} ;
```

The sampling number define the step between two measurements of the numerical probe. By default, its value is 1. Thus, the number of samples is equal to ***NumberOfSample***s modulo ***Sampling***.

**Samples = NumberOfSamples % Sampling;**

Each physical value of the probe is recorded in a standard CGNS **DataArray_t.** For instance, it is possible to store Pressure and Temperature at the probe location.

The probe can be attached to a specific Family through **FamilyName**.

A probe is basically a point local measurement. Thus, a location is optionally requested. This location is defined through a ProbeLocation_t structure. This structure is defined hereafter:

```
ProbeLocation_t <PhysicalDimension, IndexDimension> :=
{
  ProbeLocationType_t ProbeLocationType                              (r)
  DataArray_t ZonePath ;                                             (r:o)
  GridLocation_t GridLocation;                                       (r/d:o)
  IndexArray_t<IndexDimension,1> PointList;                          (r:o)

  DataArray_t<1, PhysicalDim> Coordinates ;                          (o:r)

   List( UserDefinedData_t UserDefinedData1 ... UserDefinedDataN ) ;  (o)
} ;
```

The **ProbeLocationType** can take two values: **Physical** or **GridBased.**

In the case of a **"Physical"** defined probe location, the **Coordinates DataArray_t** node should be present. It will contains the physical probe point coordinates and its size is coherent with the top Base_t Physical dimension.

In the case of a **"GridBased"** defined probe location, the **Coordinates** are not stored in the node and should be retrieved from the mesh grid. To do so, three Information are then required:

- ***ZonePath***: The path to the Zone where the **GridCoordinates** are located and from which the probe coordinates and IndexDimension can be inferred.
- ***GridLocation:*** This can be **Vertex** or **CellCenter.** If it not present, it is assumed to be **Vertex** as it is classical in CGNS. For **"Vertex",** the coordinates of the probe location are the Coordinates of the Vertex located by the indices stored in **PointList**. In the case of **CellCenter**, the coordinates of the probes correspond to a cell center. Then the indices stored in **PointList**

define the cell which cell center is computed as the probe coordinates. In this proposal, the GridLocation support is limited but if it makes sense, it can be extended later.

- ***`PointList`*****:** it stores the index of the point or cell center. It is sized according to the **`Zone`** found in **`ZonePath.`** For instance, it will be three integers for a 3D structured grid while it will be a simple index in the case of an Unstructured Zone.

## 3. Conflict and compatibility concern

No conflict are expected since we reuse existing data structure and only extend the CGNS standard with new structures.

## 4. Conclusion

This extension proposal of Monitoring and Probes provides a way for post-processing tools to have an easier way to collect and display important data of the CFD solver. It is meant to have low impact on existing CGNS SIDS structure.

## 5. Document modification list

None

# A. Appendix - Extension to the CGNS/SIDS

The previous section presented the different features needed to have a proper definition of a numerical Probe in CGNS. This section presents the modification applied to the CGNS SIDS.

To be completed.

# B. Appendix - Extension to the CGNS/MLL

To be completed

# C. Appendix – Document modification list

None