pyCGNS.PAT/Manual

Release 4.0.1

Marc Poinot

September 08, 2010

CONTENTS

The module to create and manipulate SIDS/Python trees. PAT has a *cgnslib* module with functions to create SIDS/Python compliant data structures.

CHAPTER

ONE

CGNS.PAT.CGNSLIB

A set of function to manipulate CGNS/Python nodes.

`CGNS.PAT.cgnslib.`**`newArbitraryGridMotion`**(*parent*, *name*, *valuetype='Null'*)

Returns a **new node** representing a `ArbitraryGridMotionType_t` sub-tree (chapter 11.3)

Parameters

- **parent** – CGNS/Python node
- **name** – String
- **valuetype** – String (`CGNS.PAT.cgnskeywords.ArbitraryGridMotionType`)

If a *parent* is not `None`, the **new node** is added to the parent children list. If the *parent* has already a child with name `RigidGridMotion` then only the `RigidGridMotionType` is created.

`CGNS.PAT.cgnslib.`**`newAverageInterface`**(*parent*, *valueType='Null'*)

-AverageInterface node creation -AverageInterface

'newNode:N='*newAverageInterface*'(parent:N,valueType:CG_K.AverageInterfaceType)'

Returns a new <node> representing a AverageInterface_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name AverageInterface then only the AverageInterfaceType is created. chapter 8.5.2

`CGNS.PAT.cgnslib.`**`newAxisymmetry`**(*parent*, *refpoint=*[, *0.0*, *0.0*, *0.0*], *axisvector=*[, *0.0*, *0.0*, *0.0*])

-Axisymmetry node creation -Axisymmetry

'newNode:N='*newAxisymmetry*'(parent:N,refpoint:A,axisvector:A)'

refpoint,axisvector should be a real array. Returns a new <node> representing a CG_K.Axisymmetry_t sub-tree. chapter 7.5 Add DataArray AxisymmetryAxisVector,AxisymmetryReferencePoint are required

`CGNS.PAT.cgnslib.`**`newBCData`**(*parent*, *name*)

-BCData node creation -BCData

'newNode:N='*newBCData*'(parent:N,name:S)'

Returns a new <node> representing a BCData_t sub-tree. chapter 9.5

`CGNS.PAT.cgnslib.`**`newBCDataSet`**(*parent*, *name*, *valueType='Null'*)

-BCDataSet node creation -BCDataSet

'newNode:N='*newBCDataSet*'(parent:N,name:S,valueType:CG_K.BCTypeSimple)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a BCDataSet_t sub-tree. chapter 9.4 Add node BCTypeSimple is required

`CGNS.PAT.cgnslib.`**`newBCProperty`**(*parent*, *wallfunction='Null'*, *area='Null'*)

-BCProperty node creation -BCProperty

'newNode:N='*newBCProperty*'(parent:N)'

Returns a new <node> representing a BCProperty_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 9.6

`CGNS.PAT.cgnslib.`**`newBase`**(*tree*, *name*, *ncell*, *nphys*)
-Base node creation -Base

'newNode:N='*newBase*'(parent:N,name:S,ncell:[1,2,3],nphys:[1,2,3])'

Returns a new <node> representing a CGNSBase_t sub-tree. If a parent is given, the new <node> is added to the parent children list, that is to the base list of the parent CGNSTree. Maps the 'cg_base_write' MLL chapter 6.2

`CGNS.PAT.cgnslib.`**`newBaseIterativeData`**(*parent*, *nsteps=0*, *itype='IterationValues'*)
-BaseIterativeData node creation -BaseIterativeData

'newNode:N='*newBaseIterativeData*'(parent:N,nsteps:I,itype:E)'

Returns a new <node> representing a BaseIterativeData_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 11.1.1 NumberOfSteps is required, TimeValues or IterationValues are required

`CGNS.PAT.cgnslib.`**`newBoundary`**(*parent*, *bname*, *brange*, *btype='Null'*, *family=None*, *ptype='PointRange'*)
-BC node creation -BC

'newNode:N='*newBoundary*'(parent:N,bname:S,brange:[*i],btype:S)'

Returns a new <node> representing a BC_t sub-tree. If a parent is given, the new <node> is added to the parent children list. Parent should be Zone_t, returned node is parent. If the parent has already a child name ZoneBC then only the BC_t,IndexRange_t are created. chapter 9.3 Add IndexRange_t required

`CGNS.PAT.cgnslib.`**`newCGNS`**()
-Tree node creation -Tree

'newNode:N='*newCGNS*'()'

Returns a new <node> representing a CGNS tree root. This is not a SIDS type.

`CGNS.PAT.cgnslib.`**`newChemicalKineticsModel`**(*parent*, *valueType='Null'*)
-ChemicalKineticsModel node creation -ChemicalKineticsModel

'newNode:N='*newChemicalKineticsModel*'(parent:N,valueType:CG_K.ChemicalKineticsModelType)'

Returns a new <node> representing a CG_K.ChemicalKineticsModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name ChemicalKineticsModel then only the ChemicalKineticsModelType is created. chapter 10.8 Add node ChemicalKineticsModelType is (r)

`CGNS.PAT.cgnslib.`**`newConvergenceHistory`**(*parent*, *name='GlobalConvergenceHistory'*, *iterations=0*)
-ConvergenceHistory node creation -ConvergenceHistory

'newNode:N='*newConvergenceHistory*'(parent:N,name:S,iterations:i)'

Returns a new <node> representing a ConvergenceHistory_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 12.3

`CGNS.PAT.cgnslib.`**`newCoordinates`**(*parent*, *name='GridCoordinates'*, *value=None*)
-GridCoordinates_t node creation with name GridCoordinates -Grid

'newNode:N='*newCoordinates*'(parent:N,name:S,value:A)'

Creates a new <node> representing a GridCoordinates_t sub-tree with the coordinate DataArray given as argument. This creates both the GridCoordinates_t with GridCoordinates name and DataArray_t with the argument name. Usually used to create the default grid. If the GridCoordinates_t with name GridCoordinates already exists then only the DataArray is created. If a parent is given, the new GridCoordinates_t <node> is added to the parent children list, in all cases the DataArray is child of GridCoordinates_t node. The returned node always is the DataArray_t node. chapter 7.1

CGNS.PAT.cgnslib.**newDataArray** (*parent*, *name*, *value=None*)
-DataArray node creation -Global

'newNode:N='*newDataArray*'(parent:N,name:S,value:A)'

Returns a new <node> representing a DataArray_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 5.1

CGNS.PAT.cgnslib.**newDataClass** (*parent*, *value='Userdefined'*)
-DataClass node creation -DataClass

'newNode:N='*newDataClass*'(parent:N,value:A)'

If a parent is given, the new <node> is added to the parent children list. The value argument is a DataClass enumerate. No child allowed. Returns a new <node> representing a DataClass_t sub-tree.

CGNS.PAT.cgnslib.**newDataConversion** (*parent*, *ConversionScale=1.0*, *ConversionOffset=1.0*)
-DataConversion node creation -DataConversion

'newNode:N='*newDataConversion*'(parent:N,ConversionScale:r,ConversionOffset:r)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a DataConversion_t sub-tree. chapter 5.1.1

CGNS.PAT.cgnslib.**newDescriptor** (*parent*, *name*, *value=''*)
-Descriptor node creation -Descriptor

'newNode:N='*newDescriptor*'(parent:N,name:S,text:A)'

No child allowed. Returns a new <node> representing a Descriptor_t sub-tree.

CGNS.PAT.cgnslib.**newDimensionalExponents** (*parent*, *MassExponent=0*, *LengthExponent=0*, *TimeExponent=0*, *TemperatureExponent=0*, *AngleExponent=0*)
-DimensionalExponents node creation -DimensionalExponents

'newNode:N='*newDimensionalExponents*'(parent:N,MassExponent:r,LengthExponent:r,TimeExponent:r,TemperatureExpone

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a DimensionalExponents_t sub-tree. chapter 4.4

CGNS.PAT.cgnslib.**newDimensionalUnits** (*parent*, *value=*[, *'Meter'*, *'Kelvin'*, *'Second'*, *'Radian'*, *'Kilogram'*])
-DimensionalUnits node creation -DimensionalUnits

'newNode:N='*newDimensionalUnits*'(parent:N,value=[CG_K.MassUnits,CG_K.LengthUnits,
CG_K.TimeUnits,CG_K.TemperatureUnits, CG_K.AngleUnits])'

If a parent is given, the new <node> is added to the parent children list. new <node> is composed of a set of enumeration types : MassUnits,LengthUnits, TimeUnits,TemperatureUnits,AngleUnits are required Returns a new <node> representing a DimensionalUnits_t sub-tree. chapter 4.3

CGNS.PAT.cgnslib.**newDiscreteData** (*parent*, *name*)
-DiscreteData node creation -DiscreteData

'newNode:N='*newDiscreteData*'(parent:N,name:S)'

> If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a DiscreteData_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 6.3

CGNS.PAT.cgnslib.**newEMConductivityModel** (*parent*, *valueType='Userdefined'*)
-EMConductivityModel node creation -EMConductivityModel

'newNode:N='*newEMConductivityModel*'(parent:N,valueType:CG_K.EMConductivityModelType)'

> Returns a new <node> representing a CG_K.EMConductivityModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name EMConductivityModel then only the EMConductivityModelType is created. chapter 10.9.3 Add node EMConductivityModelType is (r)

CGNS.PAT.cgnslib.**newEMElectricFieldModel**(*parent*, *valueType='Userdefined'*)
-EMElectricFieldModel node creation -EMElectricFieldModel

'newNode:N='*newEMElectricFieldModel*'(parent:N,valueType:CG_K.EMElectricFieldModelType)'

Returns a new <node> representing a CG_K.EMElectricFieldModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list.

If the parent has already a child name EMElectricFieldModel then

only the EMElectricFieldModelType is created. chapter 10.9 Add node EMElectricFieldModelType is (r)

CGNS.PAT.cgnslib.**newEMMagneticFieldModel**(*parent*, *valueType='Userdefined'*)
-EMMagneticFieldModel node creation -EMMagneticFieldModel

'newNode:N='*newEMMagneticFieldModel*'(parent:N,valueType:CG_K.EMMagneticFieldModelType)'

Returns a new <node> representing a CG_K.EMMagneticFieldModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name EMMagneticFieldModel_s then only the EMMagneticFieldModelType is created. chapter 10.9.2 Add node EMMagneticFieldModelType is (r)

CGNS.PAT.cgnslib.**newElements**(*parent*, *elementstype='Userdefined'*, *elementsconnectivity=None*, *elementsrange=None*)
-Elements node creation -Elements

'newNode:N='*newAElements*'(parent:N,elementsType:CG_K.ElementType,value:CG_K.ElementConnectivity)'

Returns a new <node> representing a Element_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name Element then only the ElementType,IndexRange_t,ElementConnectivity are created. chapter 7.3 Add node :ElementType,IndexRange_t are required

Add DataArray : ElementConnectivity is required

CGNS.PAT.cgnslib.**newFamily**(*parent*, *name*)
-Family node creation -Family

'newNode:N='*newFamily*'(parent:N,name:S)'

Returns a new <node> representing a Family_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 12.6

CGNS.PAT.cgnslib.**newFamilyBC**(*parent*, *valueType='Userdefined'*)
-FamilyBC node creation -FamilyBC

'newNode:N='*newFamilyBC*'(parent:N,valueType:CG_K.BCTypeSimple/CG_K.BCTypeCompound)'

Returns a new <node> representing a CG_K.FamilyBC_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name FamilyBC then only the BCType is created chapter 12.8 Add node BCType is required

CGNS.PAT.cgnslib.**newFlowEquationSet**(*parent*)
-FlowEquationSet node creation -FlowEquationSet

'newNode:N='*newFlowEquationSet*'(parent:N)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a CG_K.FlowEquationSet_t sub-tree. chapter 10.1

CGNS.PAT.cgnslib.**newFlowSolution**(*parent*, *name='{FlowSolution}'*, *gridlocation=None*)
-Solution node creation -Solution

'newNode:N='*newSolution*'(parent:N,name:S,gridlocation:None)'

Returns a new <node> representing a FlowSolution_t sub-tree. chapter 7.7

`CGNS.PAT.cgnslib.`**`newGasModel`**(*parent*, *valueType='Ideal'*)
-GasModel node creation -GasModel

'newNode:N='*newGasModel*'(parent:N,valueType:CG_K.GasModelType)'

> Returns a new <node> representing a CG_K.GasModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name GasModel then only the GasModelType is created. chapter 10.3 Add node GasModelType is required

`CGNS.PAT.cgnslib.`**`newGeometryReference`**(*parent*, *name='{GeometryReference}'*, *valueType='Userdefined'*)
-GeometryReference node creation -GeometryReference

'newNode:N='*newGeometryReference*'(parent:N,name:S,valueType:CG_K.GeometryFormat)'

> Returns a new <node> representing a CG_K.GeometryFormat_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name CG_K.GeometryReference then only the .GeometryFormat is created chapter 12.7 Add node CG_K.GeometryFormat_t is (r) and GeometryFile_t definition not find but is required (CAD file)

`CGNS.PAT.cgnslib.`**`newGoverningEquations`**(*parent*, *valueType='Euler'*)
-GoverningEquations node creation -GoverningEquations

'newNode:N='*newGoverningEquations*'(parent:N,valueType:CG_K.GoverningEquationsType)'

> Returns a new <node> representing a CG_K.GoverningEquations_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name GoverningEquations then only the GoverningEquationsType is created. chapter 10.2 Add node GoverningEquationsType is required

`CGNS.PAT.cgnslib.`**`newGravity`**(*parent*, *gvector=[, 0.0, 0.0, 0.0]*)
-Gravity node creation -Gravity

'newNode:N='*newGravity*'(parent:N,gvector:A)'

> Returns a new <node> representing a Gravity_t sub-tree. If a parent is given, the new <node> is added to the parent children list. gvector should be a real array chapter 12.10 Add DataArray GravityVector is required

`CGNS.PAT.cgnslib.`**`newGridConnectivity1to1`**(*parent*, *name*, *dname*, *window*, *dwindow*, *trans*)
-GridConnectivity1to1 node creation -Grid

'newNode:N='*newGridConnectivity1to1*'(parent:N,name:S,dname:S,window:[i*],dwindow:[i*],trans:[i*])'

Creates a ZoneGridConnectivity1to1_t sub-tree. If a parent is given, the new <node> is added to the parent children list, the parent should be a Zone_t. The returned node is the GridConnectivity1to1_t chapter 8.2

`CGNS.PAT.cgnslib.`**`newGridConnectivityProperty`**(*parent*)
-GridConnectivityProperty node creation -GridConnectivityProperty

'newNode:N='*newGridConnectivityProperty*'(parent:N)'

> Returns a new <node> representing a GridConnectivityProperty_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 8.5

`CGNS.PAT.cgnslib.`**`newGridCoordinates`**(*parent*, *name*)
-GridCoordinates node creation -Grid

'newNode:N='*newGridCoordinates*'(parent:N,name:S)'

Returns a new <node> representing a GridCoordinates_t sub-tree. If a parent is given, the new <node> is added to the parent children list.

`CGNS.PAT.cgnslib.`**`newGridLocation`**(*parent*, *value='CellCenter'*)
-GridLocation node creation -GridLocation

'newNode:N='*newGridLocation*'(parent:N,value:CG_K.GridLocation)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a GridLocation_t sub-tree. chapter 4.5

CGNS.PAT.cgnslib.**newIntegralData** (*parent*, *name*)
-IntegralData node creation -IntegralData

'newNode:N='*newIntegralData*'(parent:N,name:S)'

> Returns a new <node> representing a IntegralData_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 12.5

CGNS.PAT.cgnslib.**newOrdinal** (*parent*, *value=0*)
-Ordinal node creation -Ordinal

'newNode:N='*newOrdinal*'(parent:N,value=i)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a Ordinal_t sub-tree. chapter 6.3

CGNS.PAT.cgnslib.**newOversetHoles** (*parent*, *name*, *hrange*)
-OversetHoles node creation -OversetHoles

'node:N='*newOversetHoles*'(parent:N,name:S,hrange:list)'

Creates a OversetHoles_t sub-tree. the parent should be a Zone_t. If a parent is given, the new <node> is added to the parent children list. chapter 8.6 Add PointList or List(PointRange) are required

CGNS.PAT.cgnslib.**newPeriodic** (*parent*, *rotcenter=*[, *0.0*, *0.0*, *0.0*], *ratev=*[, *0.0*, *0.0*, *0.0*], *trans=*[, *0.0*, *0.0*, *0.0*])
-Periodic node creation -Periodic

'newNode:N='*newPeriodic*'(parent:N,rotcenter=A,ratev=A,trans=A)'

> Returns a new <node> representing a Periodic_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name Periodic then only the RotationCenter,RotationAngle,Translation are created. rotcenter,ratev,trans should be a real array. chapter 8.5.1 Add DataArray RotationCenter,RotationAngle,Translation are required

CGNS.PAT.cgnslib.**newPointList** (*parent*, *name='PointList'*, *value=*[ ])
-PointList node creation -PointList

'newNode:N='*newPointList*'(parent:N,name:S,value:[])'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a IndexArray_t sub-tree. chapter 4.6

CGNS.PAT.cgnslib.**newPointRange** (*parent*, *name='PointRange'*, *value=*[ ])
-PointRange node creation -PointRange

'newNode:N='*newPointRange*'(parent:N,name:S,value:[])'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a IndexRange_t sub-tree. chapter 4.7

CGNS.PAT.cgnslib.**newReferenceState** (*parent*, *name='ReferenceState'*)
-ReferenceState node creation -ReferenceState

'newNode:N='*newReferenceState*'(parent:N,name:S)'

> Returns a new <node> representing a ReferenceState_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 12.1

CGNS.PAT.cgnslib.**newRigidGridMotion** (*parent*, *name*, *valueType='Null'*, *vector=*[, *0.0*, *0.0*, *0.0*])
-RigidGridMotion node creation -RigidGridMotion

'newNode:N='*newRigidGridMotion*'(parent:N,name:S,valueType:CG_K.RigidGridMotionType,vector:A)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a CG_K.RigidGridMotion_t sub-tree. If the parent has already a child name RigidGridMotion then only the RigidGridMotionType is created and OriginLocation is created chapter 11.2 Add Node RigidGridMotionType and add DataArray OriginLocation are the only required

`CGNS.PAT.cgnslib.`**`newRind`**(*parent*, *value*)
-Rind node creation -Rind

'newNode:N='*newRind*'(parent:N,value=A)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a Rind_t sub-tree. chapter 4.8

`CGNS.PAT.cgnslib.`**`newRotatingCoordinates`**(*parent*, *rotcenter*=[, *0.0*, *0.0*, *0.0*], *ratev*=[, *0.0*, *0.0*, *0.0*])
-RotatingCoordinates node creation -RotatingCoordinates

'newNode:N='*newRotatingCoordinates*'(parent:N,rotcenter=A,ratev=A)'

Returns a new <node> representing a RotatingCoordinates_t sub-tree. If a parent is given, the new <node> is added to the parent children list. rotcenter,ratev should be a real array. chapter 7.6 Add DataArray RotationRateVector,RotationCenter are required

`CGNS.PAT.cgnslib.`**`newSimulationType`**(*parent*, *stype='NonTimeAccurate'*)
-SimulationType node creation -SimulationType

'newNode:N='*newSimulationType*'(parent:N,stype=CG_K.SimulationType)'

If a parent is given, the new <node> is added to the parent children list. Returns a new <node> representing a SimulationType_t sub-tree. chapter 6.2

`CGNS.PAT.cgnslib.`**`newThermalConductivityModel`**(*parent*, *valueType='SutherlandLaw'*)
-ThermalConductivityModel node creation -ThermalConductivityModel

'newNode:N='*newThermalConductivityModel*'(parent:N,valueType:CG_K.ThermalConductivityModelType)'

Returns a new <node> representing a CG_K.ThermalConductivityModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name ThermalConductivityModel then only the ThermalConductivityModelType is created. chapter 10.5 Add node ThermalConductivityModelType is required

`CGNS.PAT.cgnslib.`**`newThermalRelaxationModel`**(*parent*, *valueType*)
-ThermalRelaxationModel node creation -ThermalRelaxationModel

'newNode:N='*newThermalRelaxationModel*'(parent:N,valueType:CG_K.ThermalRelaxationModelType)'

Returns a new <node> representing a CG_K.ThermalRelaxationModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name ThermalRelaxationModel then only the ThermalRelaxationModelType is created. chapter 10.7 Add node ThermalRelaxationModelType is (r)

`CGNS.PAT.cgnslib.`**`newTurbulenceClosure`**(*parent*, *valueType='EddyViscosity'*)
-TurbulenceClosure node creation -TurbulenceClosure

'newNode:N='*newTurbulenceClosure*'(parent:N,valueType:CG_K.TurbulenceClosureType)'
Returns a new <node> representing a CG_K.TurbulenceClosure_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name TurbulenceClosure then only the ViscosityModelType is created. chapter 10.5 Add node TurbulenceClosureType is (r)

`CGNS.PAT.cgnslib.`**`newTurbulenceModel`**(*parent*, *valueType='OneEquation_SpalartAllmaras'*)
-TurbulenceModel node creation -TurbulenceModel

'newNode:N='*newTurbulenceModel*'(parent:N,valueType:CG_K.TurbulenceModelType)'

Returns a new <node> representing a CG_K.TurbulenceModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name TurbulenceModel then only the TurbulenceModelType is created. chapter 10.6.2 Add node TurbulenceModelType is (r)

CGNS.PAT.cgnslib.**newUserDefinedData**(*parent*, *name*)
-UserDefinedData node creation -UserDefinedData

'newNode:N='*newUserDefinedData*'(parent:N,name:S)'

> Returns a new <node> representing a UserDefinedData_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 12.9

CGNS.PAT.cgnslib.**newViscosityModel**(*parent*, *valueType='SutherlandLaw'*)
-ViscosityModel node creation -ViscosityModel

'newNode:N='*newViscosityModel*'(parent:N,valueType:CG_K.ViscosityModelType)'

> Returns a new <node> representing a CG_K.ViscosityModel_t sub-tree. If a parent is given, the new <node> is added to the parent children list. If the parent has already a child name ViscosityModel then only the ViscosityModelType is created. chapter 10.4 Add node ViscosityModelType is (r)

CGNS.PAT.cgnslib.**newZone**(*parent*, *name*, *size=(2, 2, 2)*, *ztype='Structured'*, *family=''*)
-Zone node creation -Zone

'newNode:N='*newZone*'(parent:N,name:S,size:(I*),ztype:CG_K.ZoneType)'

Returns a new <node> representing a Zone_t sub-tree. If a parent is given, the new <node> is added to the parent children list. Maps the 'cg_zone_write' MLL chapter 6.3

CGNS.PAT.cgnslib.**newZoneGridConnectivity**(*parent*, *name*, *ctype='Null'*, *donor=''*)
-GridConnectivity node creation -Grid

'newNode:N='*newZoneGridConnectivity*'(parent:N,name:S,ctype:S)'

Creates a ZoneGridConnectivity_t sub-tree with a sub-node depending on the type of connectivity. This sub-node is returned. If a parent is given, the new <node> is added to the parent children list, the parent should be a Zone_t. chapter 8.1

CGNS.PAT.cgnslib.**newZoneIterativeData**(*parent*, *name*)
-ZoneIterativeData node creation -ZoneIterativeData

'newNode:N='*newZoneIterativeData*'(parent:N,name:S)'

> Returns a new <node> representing a ZoneIterativeData_t sub-tree. If a parent is given, the new <node> is added to the parent children list. chapter 11.1.2

CHAPTER

TWO

CGNS.PAT.SIDS

All the CGNS/SIDS structures using CGNS.PAT as API.

CHAPTER

THREE

CGNS.PAT.CGNSKEYWORDS

CGNS.PAT.**cgnskeywords**

CHAPTER

FOUR

CGNS.PAT.CGNSERRORS

CGNS.PAT.**cgnserrors**

PYTHON MODULE INDEX