CNRS Autrans 2010

Sphinx

Marc Poinot
Numerical Simulation
For
Aerodynamics and Aeroacoustics Dept.

ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

Sphinx

- Géneration de documentation
  - Basé sur ReST
  - Production de documents html, LaTeX...
  - Indépendance document/ support de production
  - Et surtout c'est super trop beau ! *(voire hyper top cool !)*
- Installation
  - Module Python standard
  - Pas mal de dépendances...
- Utilisation
  - Texte ASCII dans le source ou dans des fichiers séparés
  - Production simple
  - Nombreux modules externes

Mon Module - Page Maison

CGNS.NAV

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

```
CGNS.NAV
```

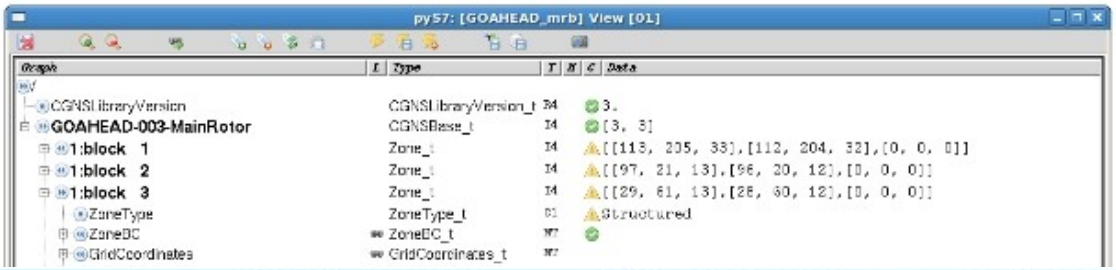
and the pyCGNS browser appears. It can load/save CGNS files with the HDF5/ADF and Python formats, parse and display the contents, edit the contents using simple edit and copy/paste commands, select nodes using complex queries, use already defined patterns such as the *SIDS* patterns...

- QuickStart
- Tree View
- Option View
  - Options
- Pattern View
- 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 sreenshot below shows a `CGNS.NAV` with various windows.



Next topic

QuickStart

This Page

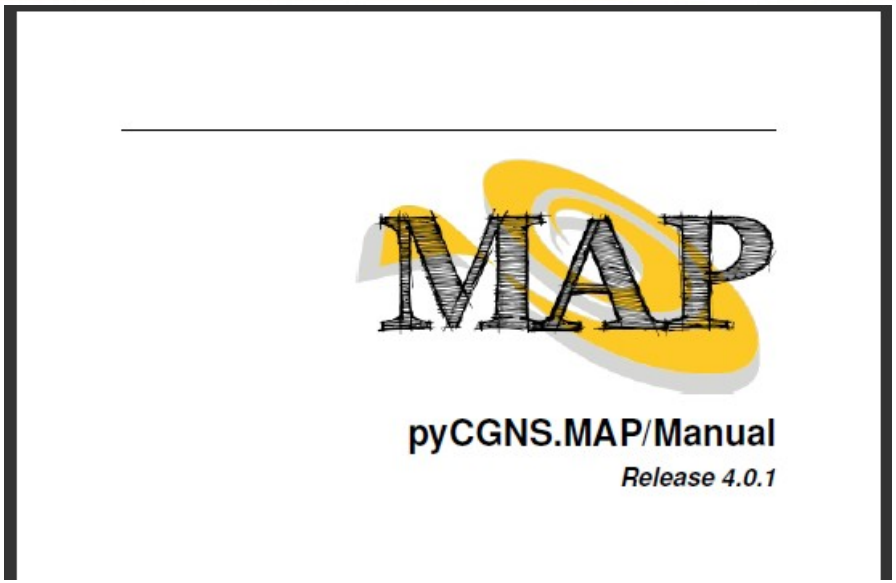
Show Source

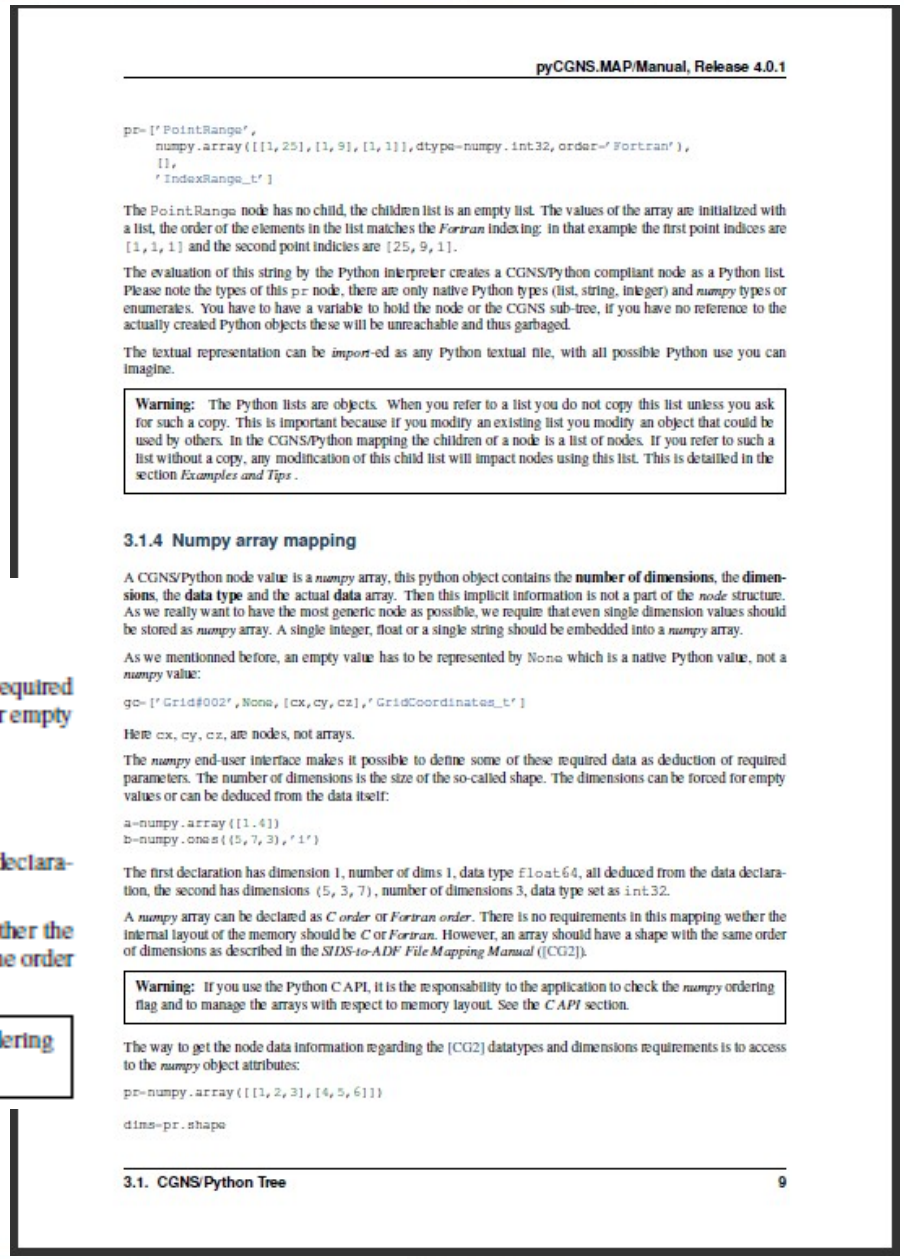
Quick search

Go

Enter search terms or a module, class or function name.

Mon Module - LaTeX






```
pr=['PointRange',
    numpy.array([[1,25],[1,9],[1,1]],dtype=numpy.int32,order='Fortran'),
    [],
    'IndexRange_t']
```

The PointRange node has no child, the children list is an empty list. The values of the array are initialized with a list, the order of the elements in the list matches the *Fortran* indexing: in that example the first point indices are `[1,1,1]` and the second point indicies are `[25,9,1]`.

The evaluation of this string by the Python interpreter creates a CGNS/Python compliant node as a Python list. Please note the types of this `pr` node, there are only native Python types (list, string, integer) and *numpy* types or enumerates. You have to have a variable to hold the node or the CGNS sub-tree, if you have no reference to the actually created Python objects these will be unreachable and thus garbaged.

The textual representation can be *import*-ed as any Python textual file, with all possible Python use you can imagine.

Warning: The Python lists are objects. When you refer to a list you do not copy this list unless you ask for such a copy. This is important because if you modify an existing list you modify an object that could be used by others. In the CGNS/Python mapping the children of a node is a list of nodes. If you refer to such a list without a copy, any modification of this child list will impact nodes using this list. This is detailed in the section *Examples and Tips* .

3.1.4 Numpy array mapping

A CGNS/Python node value is a *numpy* array, this python object contains the **number of dimensions**, the **dimensions**, the **data type** and the actual **data** array. Then this implicit information is not a part of the *node* structure. As we really want to have the most generic node as possible, we require that even single dimension values should be stored as *numpy* array. A single integer, float or a single string should be embedded into a *numpy* array.

As we mentionned before, an empty value has to be represented by `None` which is a native Python value, not a *numpy* value:

```
gc=['Grid#002',None,[cx,cy,cz],'GridCoordinates_t']
```

Here `cx`, `cy`, `cz`, are nodes, not arrays.

The *numpy* end-user interface makes it possible to define some of these required data as deduction of required parameters. The number of dimensions is the size of the so-called shape. The dimensions can be forced for empty values or can be deduced from the data itself:

```
a=numpy.array([1.4])
b=numpy.ones((5,7,3),'i')
```

The first declaration has dimension 1, number of dims 1, data type `float64`, all deduced from the data declaration, the second has dimensions `(5,3,7)`, number of dimensions 3, data type set as `int32`.

A *numpy* array can be declared as *C order* or *Fortran order*. There is no requirements in this mapping wether the internal layout of the memory should be *C* or *Fortran*. However, an array should have a shape with the same order of dimensions as described in the *SIDS-to-ADF File Mapping Manual* ([CG2]).

Warning: If you use the Python C API, it is the responsability to the application to check the *numpy* ordering flag and to manage the arrays with respect to memory layout. See the *C API* section.

The way to get the node data information regarding the [CG2] datatypes and dimensions requirements is to access to the *numpy* object attributes:

```
pr=numpy.array([[1,2,3],[4,5,6]])

dims=pr.shape
```

Mise en oeuvre - 1

- Documenter
  - Construire une arborescence documentaire
  - Documenter son source Python
  - Ecrire des fichiers .txt
- Configurer
  - Définir un répertoire de production
  - Faire le fichier de configuration
- Produire
  - Faire un script de lancement
  - Résultats:
  - Un répertoire html autonome tarzipable
    - Accepté par SourceForge, www.python-science.org, ...
  - Un répertoire LaTeX et un PDF (avec pdflatex)

Mise en oeuvre - 2

- Production
  - Script de lancement
    - Makefile, setup.py, script shell...

```
# should be run in root dir (i.e. setup.py as brother)
export RDIR=./.scons.linux2.tmp/build

mkdir -p $RDIR/doc/html
mkdir -p $RDIR/doc/html/images
cp $RDIR/src/include/CHLone/*.txt ./doc

sphinx-build -c doc -b html doc $RDIR/doc/html
sphinx-build -c doc -b latex doc $RDIR/doc/latex

cp doc/images/* $RDIR/doc/html/images

(cd $RDIR/doc/html; tar cvf ../../../CHLone-html.tar .)
(cd $RDIR/doc/latex; pdflatex CHLone.tex)
```

ReST - 1

Restructured Text

- Les commandes sont dans le corps du texte
  - Le texte est en ASCII
    - Le formatage du texte reste très simplifié
    - N'imaginez pas faire de formules de Math
- Dialecte
  - De préférence se reporter à la documentation Sphinx
    - Tout ReST ne marche pas
    - Certaines syntaxes propres à Sphinx
  - Quelques éléments

```
*italique*
**gras**
``courier``
Liste d'items
- item 1
- item 2
```

ReST - 2

Titres

- L'outil identifie les niveaux de titres au fur et à mesure qu'il les rencontre
  - Un titre est un texte souligné (avec le bon nombre de soulignés...)
  - Vous définissez arbitrairement les caractères utilisés

```
About pyCGNS
++++++++++++

Package contents
----------------

NAV depends
~~~~~~~~~~~
```

Directives - 1

Table des matières

- La directive *toctree*
  - Définit votre arborescence
    - Une entrée est un fichier .txt
    - Paramètres divers de profondeur, renomage...
  - Génére une table des matières
    - Ainsi qu'un bandeau latéral pour la navigation HTML

```
.. toctree::
   :maxdepth: 2

   install
   interfaces
   implementation
   ../src/readme
   ../test/readme
```

Directives - 2

- Liens
  - Des directives faciles
    - Liens externes

```
`CHLone` is released under the
`LGPL v2 license <http://www.gnu.org/licenses/lgpl-2.1.txt>`_ and is
available on `SourceForge <http://sourceforge.net/projects/chlone>`_.
```

  - Liens internes
    - Une balise définit un point de référencement

```
You can find :ref:`here <releases>`_ a more detailled release note for
this version and previous release notes.
```

  - Balise pour le lien interne

```
.. _releases:
Current available releases and associated notes
```

Directives - 3

Insertion de code

- Colorisation syntaxique automatique

```
the ``>>>`` string is the python interpreter prompt)::

>>>import CGNS.MAP
>>>(tree,links)=CGNS.MAP.load("./Disk.hdf",CGNS.MAP.S2P_FOLLOWLINKS)
```

Index

- Création d'entrées
- Génération automatique
  - Si demandée...

```
.. index::
   single: CHLone
   pair: Installation; Python
   pair: Installation; numpy
   pair: Installation; hdf5
   pair: Installation; CHLone
```

Directives - 4

Images

- Fonctionne en HTML et LaTeX
- Le format (extension fichier) est détecté suivant le mode de production
- Prise en compte de jpeg et png

```
.. image:: ../doc/images/options_view_1.png
   :width: 16cm
   :align: center
```

Commentaires

- Deux fois un point suivi de tout sauf une directive...

```
.. ------------------------------------------------------------
.. pyCGNS - CFD General Notation System -
.. See license.txt in the root directory of this Python module source
.. ------------------------------------------------------------
```

Directives - 5

- Boites de warning

```
.. warning::

  You don't need *libcgns* for the *CGNS/Python* mapping.
```

- Notes de bas de page
  - La note elle-même

```
.. [#n5] The *pyCGNS* Python module defines services on the top
         of this CGNS/Python mapping.
```

  - La référence à la note

```
The mapping here is *NOT* a library [#n5]_, it is the lowest possible
```

Directives - 6

Tables

- Pas extraordinaire...
- Très difficile à éditer
- ASCII art
- Préférer un générateur

```
+-------------+-----------------+-----------+
|*ADF type*   |*Numpy type(s)*  |*Remarks*  |
+=============+=================+===========+
|`I4`         |`'i' int32`      | \(1)      |
+-------------+-----------------+-----------+
|`I8`         |`'l' int64`      | \(2)      |
+-------------+-----------------+-----------+
|`R4`         |`'f' float32`    | \(3)      |
+-------------+-----------------+-----------+
|`R8`         |`'d' float64`    | \(4)      |
+-------------+-----------------+-----------+
|`C1`         |`'c' '\|S1'`     | \(5)      |
+-------------+-----------------+-----------+
```

Directives - 7

- Spécifiques au HTML
  - Sont visibles seulement pour cette production
  - Permet de placer du code HTML brut
  - Autorise la mise en oeuvre de scripts Java
    - Nécessite alors des templates personnalisés

```
this is plain text used for all prods
raw:: html
  <a aref="http://www.python.org">Python</a>
plain text...
```

Autodoc

Documentation embarquée

- Dans le module
  - Texte en triples-quotes
  - Associé à l'élément syntaxique Python précédent
  - Permet de palier au manque de définition d'interface de Python
- Deux aspects
  - La génération automatique

```
CGNS.WRA.wrapper
----------------
The CGNS/MLL library wrapper.

.. automodule:: CGNS.WRA.wrapper
   :members:
```

  - La référence

```
:py:func `CGNS.WRA.proxy`
```

Langages C/C++/Fortran

- Génération externe
  - Notion de *domain*
    - Espace de références associé à un contexte
      - Python, C, C++, Java...
  - Nécessité de faire un outil externe produisant le texte
    - Avec des scripts Python
    - Avec des générateurs/parsers existants
- Attention
  - Sphinx sait identifier des éléments syntaxiques du C
  - Mais il ne sait **pas** produire ces éléments syntaxiques

Fichier de configuration - 1

- Fichier Python
  - Utilisation des variables et modules disponibles
- Une partie générique
  - Localisation des sources
    - Exclusions, inclusions de répertoires
    - Extensions, encodage...
  - Identification du module
    - Projet, auteur, date, version...
- Une partie par mode de production
  - HTML
    - Répertoires contenant les images
    - Options d'apparence
  - LaTeX
    - Format de papier, polices de caractères...

Fichier de configuration - 2

```
import os
# -- General configuration --
source_suffix = '.txt'
master_doc = 'index'
project = u'CHLone'
copyright = u'2010, Marc Poinot'
version = '0.4'
exclude_trees = [os.environ['RDIR']]
pygments_style = 'sphinx'
# -- Options for HTML output --
html_theme = 'sphinxdoc'
html_short_title = 'CHLone'
html_logo = 'images/CHLone-logo.jpg'
html_use_index = True
# -- Options for LaTeX output --
latex_paper_size = 'a4'
latex_font_size = '10pt'
latex_documents = [ ('index', 'CHLone.tex', u'CGNS/HDF5 Implementation',
                     u'Marc Poinot', 'manual'),]
latex_logo = 'images/CHLone-frontpage.jpg'
```

HTML - Encadrés

- full int32/int64/float32/float64 support
- versatile query system
- better support for restrictions as well as for extensions
- optional checks (also remove useless extra checks such as has_att)
- pattern-based functions identification
- HDF5 physical driver independence
- test suites with coverage

Note

CHLone stands for *CGNS HDF5 Library open to numerous extensions*.

Release status

CHLone is released under the LGPL v2 license and is available on SourceForge.

The latest *CHLone* release is v0.4, the most important news are:

- **Links** are managed with an input specification as well as an output result.
- **Errors** are changed to be independant of the interface level.
- **Documentation** uses now *Sphinx*.

Warning

The *L3* interface is the only interface of *CHLone* that is safe for your application. The *C1* and the *A2* interfaces are only prototypes which should not be used.

You can find here a more detailled release note for this version and previous release notes.

HTML - Table des matières

- Installation
  - HDF5 requirements
  - Build and install
- Interfaces
  - L3 interface
  - A2 interface
  - C1 interface
  - Demos
- Implementation
  - Transactional features
  - Fortran vs C ordering in data arrays
  - MPI-based parallel applications
  - Multi-threading applications
  - Link traversal policy
  - Tests

HTML - Index

Index

C | H | I | N | P

C

CGNS/ADF
CGNS/MLL
  Installation
	CHLone
	  Installation

H

hdf5
  Installation

I

Installation
  CGNS/MLL
  CHLone
  Python
  hdf5
  numpy
  pyCGNSconfig_user.py

N

PyCGNS

Quick search

Go

Enter search terms or a module, class or function name.

HTML - Notes

Footnotes

[1] *(1, 2)* The *SIDS-to-ADF* [CG2] or *SIDS-toHDF5* [CG3] documents of *cgns.org* have the detailled description of each node of the standard.

[2] We show in the textual representation section that this dimension ordering could lead to quite complicated Python code, but our choice was to take the implementation from the 'Fortran' world, which is the basis of the CFD world. It would be up to the user to write his own application layer for a better Python interface.

[3] A Python list is a reference, if you put a list as a child of another list the Python interpreter actually refers to the child list. Then a child can be shared by two different lists if you do not ask for a copy. In other words, the links are the natural way of referencing to lists in Python.

[4] The No-class paradigm: One would ask why we do not have a Python class or module that would implement the CGNS/Python mapping. For example, we could define a class like:

```
n=CGNSnode()
n.name="PointRange"
n.value=[[1,2,3],[4,5,6]]
n.type="IndexRange_t"
```

The answer is that we do *not* want to force people to use a specific implementation for this mapping.

The object-oriented paradigm leads to have a hierarchy of types (classes) resulting from the factorization of the actual values found in the application. In our case, we do not want such a

HTML - Tables

Flags

The flags are integers that can be OR-ed or XOR-ed to set/unset specific behavior during the load and the save. The boolean operators are used for the flag settings:

```
flags=CGNS.MAP.S2P_FOLLOWLINKS|CGNS.MAP.S2P_TRACE

flags =flags&~CGNS.MAP.S2P_TRACE
flags&=~CGNS.MAP.S2P_TRACE
```

The table below gives the *CGNS.MAP* flags.

Flag variable	*Function*
S2P_NONE	Clear all flags, set to zero.
S2P_ALL	Set all flags, set to one.
S2P_TRACE	Set the trace on, messages are sent to 'stdout'
S2P_FOLLOWLINKS	Continue to parse the linked-to tree (1)
S2P_MERGELINKS	Forget all link specifications. (2)
S2P_COMPRESS	Sets the compress flag for 'DataArray_t' (2)
S2P_NOTRANSPOSE	No *dimensions* transpose during load and save. (5)
S2P_NOOWNDATA	Forces the *numpy* flag \~NPY_OWNDATA (1) (3)
S2P_NODATA	Do not load large 'DataArray_t' (2) (4)
S2P_UPDATE	not used
S2P_DELETEMISSING	not used

There is no requirements or check on which flag can or cannot be associated with another flag.

Remarks:

1. Only when you are *loading* a tree.
2. Only when you are *saving* a tree.
3. Which means all DataArray_t actual memory zones will **NOT** be released by Python.
4. The term *Large* has to be defined. The *save* will **NOT** check if the CGNS/Python tree was

HTML - Code source

removes it from the list:

```
A2_ArgList_t *A;
A=A2_argNew(C);
A2_argPut(C,A,Name_s,A2E_String,"{Base#001}");
```

The type of the argument is an enumerate, the name can be any string or one of the CGNS/SIDS recommanded names. If you put twice the same name, the type dans the value are changed in the existing pair with the same name key. Then a name can appear once in a list. Once you have used the arguments, you delete the complete list:

```
A2_argPut(C,A,Name_s,A2E_String,"{Base#001}");
A2_argPut(C,A,CellDimension_s,A2E_Integer,3);
A2_argPut(C,A,PhysicalDimension_s,A2E_Integer,3);
...
A2_argDel(C,A);
```

The get looks for the required name and returned the value in the variable you give. The get should have the same name/data type as the put:

```
char *name;
A2_argGet(C,A,Name_s,A2E_String,&name);
```

Queries

The queries are used to explore a CGNS tree and retrieve a set of HDF5 nodes from a set of criteria. The query is built using `A2_queNew` and `A2_quePut`. The `A2_select` runs the query and returns a list of `hid_t` matching the criteria:

```
Q=A2_queNew(C);
A2_quePut(C,Q,A2Q_OR,A2Q_MatchName,"{Zone#001}");
A2_quePut(C,Q,A2Q_OR,A2Q_MatchName,"{Zone#002}");
```

Sphinx - Page Maison

SPHINX

PYTHON DOCUMENTATION GENERATOR

Sphinx home | Documentation » modules | index

Welcome

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation, written by Georg Brandl and licensed under the BSD license.

What users say:

"Cheers for a great tool that actually makes programmers **want** to write documentation!"

It was originally created for the new Python documentation, and it has excellent facilities for the documentation of Python projects, but C/C++ is already supported as well, and it is planned to add special support for other languages as well. Of course, this site is also created from reStructuredText sources using Sphinx!

Sphinx is under constant development. The following features are present, work fine and can be seen "in action" in the Python docs:

- **Output formats**: HTML (including Windows HTML Help), LaTeX (for printable PDF versions), manual pages, plain text
- **Extensive cross-references**: semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- **Hierarchical structure**: easy definition of a document tree, with automatic links to siblings, parents and children

A project

Download

Current version: **1.0.4**

Get Sphinx from the Python Package Index, or install it with:

```
easy_install -U Sphinx
```

Latest development version docs are also available.

Questions? Suggestions?

Join the Google group:

your@email
Subscribe

or come to the `#pocoo` channel