

# *NPTool Documentation*

Adrien MATTA

April 3, 2009

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>NPSimulation</b>	<b>3</b>
2.1	The way it's work . . . . .	3
2.2	Adding a detector to NPS . . . . .	3

# 1 Introduction

NPTool, Nuclear Physics Tool, aim to be a coherent set of programm usefull for Nuclear Physicist, especially those studying structure experimentally.

Because each experiment is differents, people get used to exchange code and modified it to their needs. What NPT do is provinding a common platform, each user can then add its own class to fit its own needs and share those class with our community. Working this way, each contribution is valorised(?).

## 2 NPSimulation

NPSimulation is build on top of Geant4. It's provide a coherent and modular sets of classes that can be easily modified for your purpose.

### 2.1 The way it's work

Because NPS is build on top of Geant4, you need C++ knowledge and Geant4 skills to understand how NPS work. NPS is a build as a modular basis that fit Nuclear Physicist needs:

- Generate Nuclear Physics Event such as transfert
- Deal with different Detectors and Configuration (Number of module, positionning...)

### 2.2 Adding a detector to NPS

What you need is to create a new classes to your detector. In the following line, you can replace myDetector by your detector name. First create a new file myDetector.hh in the include directory of NPS.

Here is the layout of your header file :

```
#ifndef MyDetector_H
#define MyDetector_H
#include "VDetector.hh"

class myDetector : public VDetector
{
    public:          // Creator and Destructor
        myDetector() ;
        ~myDetector() ;
}
```

```

public:      // Inherit from VDetector Class
    void ReadConfiguration(string)
    void ConstructDetector(G4LogicalVolume*)
    void AddDetectorToTree(TTree*)
    void ReadSensitive(const G4Event*)

public:      // Specific methods of myDetector
    void MySpecificMethod ()      ;

private:    // Private member of myDetector
    TTree* m_Tree                ;
    // Optional but recommended:
    myDetectorData m_Data        ;

private:    // Private member of myDetector
    G4double OtherParameter      ;
};
#endif

```

One can not that four methods come from the VDetector class. In VDetector class, those method are declared as virtual, and not implemented. You must implement those method in your class or the compilation will failed.