

GammaLib - Naming_conventions - # 2

{{lastupdated_at}} by {{lastupdated_by}}

Naming conventions

Strict naming conventions should be followed throughout GammaLib to make the library coherent, easy to use, and non-conflicting with other packages.

Classes

GammaLib classes **shall** start with an upper case G, followed by a human understandable name of which the first letter is in upper case. Examples are

```
GEnergy  
GModels  
GEventBin
```

Derived classes names **should** be built by prepending a specification attribute to the base class name, e.g.

```
GEvent => GEventBin  
GFitsImage => GFitsImageDouble
```

Functions

GammaLib functions names **should** be as close as possible to functions names used commonly in C++. For example, to take the sine of every element of a GammaLib vector, the `sin` name is used:

```
GVector elements;  
...  
GVector sin_of_elements = gammalib::sin(elements);
```

This implies that GammaLib functions need to live in a proper namespace. The GammaLib namespace for functions is `gammalib`.

Global variables

Global variables in GammaLib **should** start with `g_`. The rest of the name should be as descriptive and specific as possible.

Constants

Constants used throughout GammaLib **shall** also be placed in the `gammalib` namespace.

Using namespaces

To avoid naming conflicts, GammaLib **shall** never use internally any using directive, but **shall** always put the namespace in front of the function or constant.