

GammaLib - Development_Notes - # 41

{{lastupdated_at}} by {{lastupdated_by}}

Development Notes

This page summarizes development notes for the GammaLib software.

The page is organized following the module structure of GammaLib. Please use these pages to track your developments and any information related to your coding. For example, note the basic assumptions that have been made in implementing the code. Or note a kluge that was used. Also, specify if things need to be improved further.

If you add a new page, please put

{{lastupdated_at}} by {{lastupdated_by}}

in the first line of the Wiki page so that the modification date and the person who did the modification is visible at top of the page.

Introduction

As introduction, here are a number of useful links and pages that introduce the style and methods that are used for the GammaLib developments.

- [Coding and Design Conventions](#)
- [\[\[Coding principles\]\]](#)
- [\[\[Coding techniques\]\]](#)
- [C++ Style and Technique FAQ](#)

General considerations

- [\[\[Naming conventions\]\]](#)
- [\[\[Exceptions\]\]](#)
- [\[\[Printing\]\]](#)

Base classes (base)

- [\[\[GBase\]\]](#) - GammaLib interface class
- [\[\[GRegistry\]\]](#) - Registry interface class
- [\[\[GContainer\]\]](#) - Container interface class

High-level analysis support

Observation handling (obs)

- [\[\[GObservations\]\]](#) - Observation container
- [\[\[GObservation\]\]](#) - Abstract observation base class
- [\[\[GTime\]\]](#) - GammaLib time representation

Model handling (model)

- [\[\[GModels\]\]](#) - Model container
- [\[\[GModel\]\]](#) - Abstract model base class
- [\[\[GModelSky\]\]](#) - Sky models
- [\[\[GModelSpectralExpPlaw\]\]](#) - Exponential power law spectrum
- [\[\[GModelSpectralGauss\]\]](#) - Gaussian (for lines)
- [\[\[GModelSpectralNodes\]\]](#) - Node function
- [\[\[GModelSpectralLogParabola\]\]](#) - Log parabola spectrum
- [\[\[GModelSpectralPlaw\]\]](#) - Power law spectrum (defined by intensity at pivot energy)
- [\[\[GModelSpectralPlaw2\]\]](#) - Power law spectrum (defined by integral flux)

Skymap handling (sky)

- [[GRegion]] - Region handling
- [[GSkyDir]] - Sky direction handling

Application handling (app)

Instrument specific modules (inst)

- [[CTA interface]] (cta)
- [[Fermi/LAT interface]] (lat)
- [[CGRO/COMPTEL interface]] (com)
- [[COSI interface]] (cos)
- [[INTEGRAL/SPI interface]] (spi)
- [[Multi-wavelength interface]] (mwl)

Core services

Numerics (numerics)

Linear algebra (linalg)

Optimizers (opt)

- [[GOptimizerLM]] - Levenberg Marquardt optimizer

Support functions (support)

Interfaces

FITS interface (fits)

XML interface (xml)

Virtual Observatory interface (vo)

Test suites (test)

- [[Unit testing]]

Python bindings

- [[Python 2.x versus Python 3.x]]