

## GammaLib - GRegion - # 6

{{lastupdated\_at}} by {{lastupdated\_by}}

# GRegion

## GRegion class

The GRegion class is an abstract virtual base class that describes a region on the sky in arbitrary coordinates. Here a draft of the class definition:

```
class GRegion {
    virtual void    clear(void) = 0;
    virtual GRegion* clone(void) const = 0;
    virtual bool    isin(const GSkyDir& dir) const = 0;
    virtual void    read(const GXmlElement& xml) = 0;
    virtual void    write(GXmlElement& xml) const = 0;
    virtual std::string print(void) const = 0;
}
```

## GRegions class

The GRegions class is a container to hold derived classes of type GRegion. Here a draft of the class definition:

```
class GRegions {
    GRegion&    operator[](const int& index);
    const GRegion& operator[](const int& index) const;
    void        clear(void);
    int         size(void) const;
    void        append(const GRegion& region, const bool& include = true);
    void        insert(const int& index, const GRegion& region, const bool& include = true);
    void        extend(const GRegions& regions);
    void        pop(const int& index = -1);
    void        load(const std::string& filename);
    void        save(const std::string& filename) const;
    void        read(const GXmlElement& xml);
    void        write(GXmlElement& xml) const;
    bool        isin(const GSkyDir& dir) const = 0;
    std::string print(void) const;
}
```

Note that the append, insert, extend and pop methods are standard methods for container classes (although insert, extend and pop is so far rarely implemented in GammaLib). The load and save methods should act on XML files, the read and write methods on GXmlElement objects (which are basically opened XML files). We may also implement methods for loading and saving ds9 region files (e.g. load\_ds9 and save\_ds9).

The GRegions container could then be used as follows to select pixels from a sky map:

```
GRegions regions("my_preferred_regions.xml");
GSkymap map("my_nice_sky_map.fits");
GSkymap selected = map.select(regions);
selected.save("my_selected_pixels.fits");
```

In the same way it can be used internally by the CTA event cube class, as the event cube data are stored in a GSkymap object.

## Region XML format

We can of course invent whatever format we like, but it would be worth checking if some XML format exists already for regions, e.g. in the virtual observatory.