

GammaLib - Feature #1602

Create GFitsExtension class

12/13/2015 01:37 AM - Knödseder Jürgen

Status:	New	Start date:	01/18/2016
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
Create a GFitsExtension class to handle FITS extension naming, numbering and versioning in a uniform way.			
The class should copy the actual extension handling from GFilename so that FITS file extensions can be accessed by passing a GFitsExtension object. Automatic type conversion should be provided to directory provide a string.			

History

#1 - 01/11/2016 10:15 PM - Knödseder Jürgen

- Release set to *gammalib-1.1.0*

#2 - 01/18/2016 03:24 PM - Knödseder Jürgen

- Target version set to *1.1.0*

- Start date set to *01/18/2016*

#3 - 01/19/2016 12:05 AM - Knödseder Jürgen

We should think about supporting multiple extensions also. This is needed for the GCTAEventList class that writes two extensions into a FITS file (an event extension and a Good Time Interval extension).

A possible format for this could be:

```
test_cta_events1.fits[EVENTS2;GTI2]
```

and with extension versions

```
test_cta_events1.fits[EVENTS2,1;GTI2,2]
```

If we decide to support this then GFitsExtension should be able to deal with an arbitrary number of extensions.

#4 - 01/19/2016 12:08 AM - Knödseder Jürgen

Note that instead of generally supporting this we could have two classes: GFitsExtension and GFitsExtensions. The second would be a container of

GFitsExtension, but it would have a constructor that parses the multiple-extension string and creates instances of GFitsExtension.

#5 - 02/18/2016 06:14 PM - Knödlseider Jürgen

Here a possible code simplification that should be reached with the GFitsExtension class. This code snippet is from GSKyMap::load:

```
// If an extension name is specified then first try loading that
// extension
if (filename.has_extname()) {
    const GFitsHDU& hdu = *fits.at(filename.extname());
    if (is_healpix(hdu)) {
        read_healpix(static_cast<const GFitsTable&>(hdu));
        loaded = true;
    }
    else if (is_wcs(hdu)) {
        read_wcs(static_cast<const GFitsImage&>(hdu));
        loaded = true;
    }
}

// ... otherwise is an extension number is specified then try loading
// the corresponding HDU
else if (filename.has_extno()) {
    const GFitsHDU& hdu = *fits.at(filename.extno());
    if (is_healpix(hdu)) {
        read_healpix(static_cast<const GFitsTable&>(hdu));
        loaded = true;
    }
    else if (is_wcs(hdu)) {
        read_wcs(static_cast<const GFitsImage&>(hdu));
        loaded = true;
    }
}
```

should become

```
if (filename.has_extension()) {
    const GFitsHDU& hdu = *fits.at(filename.extension());
    if (is_healpix(hdu)) {
        read_healpix(static_cast<const GFitsTable&>(hdu));
        loaded = true;
    }
    else if (is_wcs(hdu)) {
        read_wcs(static_cast<const GFitsImage&>(hdu));
        loaded = true;
    }
}
```

Note that the latter case would also handle extension versions, while the actual code does not handle this.

#6 - 06/21/2016 05:24 PM - Knödseder Jürgen

- Target version deleted (1.1.0)

#7 - 06/21/2016 10:03 PM - Knödseder Jürgen

- Target version set to 1.2.0

#8 - 03/03/2017 10:16 AM - Knödseder Jürgen

- Target version changed from 1.2.0 to 1.3.0

#9 - 06/06/2017 10:29 PM - Knödseder Jürgen

- Target version deleted (1.3.0)