

## GammaLib - Feature #800

### Implement GUrl class for transparent handling of URL operations

03/14/2013 08:15 AM - Knödlseeder Jürgen

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assigned To:</b>	Knödlseeder Jürgen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	00-08-00		

#### Description

GammaLib should be handle to perform the same operations on files, text strings and distant media (accessed for example via http, ftp, etc.). This can be achieved by implementing an abstract base class GUrl that implements the following basic operations:

- open for opening (e.g. file opening, socket opening, string clearing, etc.)
- close for closing (e.g. file closing, socket closing, etc.)
- read reading of the media
- write writing to the media
- getc getting a character from the media
- putc putting a character into the media

Derived classes are then implemented to handle the various media:

- GUrlString handles a simple string
- GUrlFile handles a flat file
- GUrlHttp handles a network connection using http
- GUrlHttps handles a secured network connection (need to check whether this should be merged with GUrlHttp)
- GUrlFtp handles ftp connections

#### History

##### #1 - 03/14/2013 08:20 AM - Knödlseeder Jürgen

- Description updated

- Priority changed from Normal to High

##### #2 - 03/14/2013 06:12 PM - Knödlseeder Jürgen

- Target version set to 00-08-00

##### #3 - 03/14/2013 06:13 PM - Knödlseeder Jürgen

- Subject changed from Implement GUrl class for transparent handling of URL handling operations to Implement GUrl class for transparent handling of URL operations

##### #4 - 03/14/2013 11:56 PM - Knödlseeder Jürgen

- % Done changed from 0 to 80

The GUrl abstract base class has been implemented that defines the standard interface for the URL classes:

```
class GUrl : public GBase {
public:
    // Constructors and destructors
    GUrl(void);
    GUrl(const GUrl& url);
    virtual ~GUrl(void);

    // Operators
    GUrl& operator=(const GUrl& url);

    // Pure virtual methods
```

```

virtual void    clear(void) = 0;
virtual GUri*  clone(void) const = 0;
virtual void    open(const std::string& url, const std::string& mode) = 0;
virtual void    close(void) = 0;
virtual int     read(void* buffer, const int& nbyte) = 0;
virtual int     write(const void* buffer, const int& nbyte) = 0;
virtual int     getchar(void) = 0;
virtual void    putchar(const int& character) = 0;
virtual void    scanf(const char* format, ...) = 0;
virtual void    printf(const char* format, ...) = 0;
virtual std::string print(void) const = 0;
};

```

The GUriFile class has been implemented that handles flat files. Unit tests have been added for C++ and Python to validate the class. Note that the Python interface differs slightly from the C++ interface. In particular, the `scanf()` and `printf()` methods are so far missing. We may see later how those could be implemented. I guess this requires some deep SWIG coding.

#### **#5 - 03/14/2013 11:56 PM - Knödseder Jürgen**

- Status changed from New to In Progress

#### **#6 - 03/15/2013 10:10 AM - Knödseder Jürgen**

- Status changed from In Progress to Feedback

- % Done changed from 80 to 100

The class GUriString has been added for reading and writing to strings. This completes the functionality that is needed for the moment, hence the feature can be closed.

Note that the `printf()` method has an internal buffer limitation and the `scanf()` method does not advance the internal pointer, hence `printf()` can be used with limitations and `scanf()` cannot really be used.

See #801.

#### **#7 - 12/14/2013 01:00 AM - Knödseder Jürgen**

- Status changed from Feedback to Closed

- Assigned To set to Knödseder Jürgen