GammaLib - Bug #2917

GTime.utc(string) rounds up seconds above 59.5 to the next minute

06/26/2019 05:32 PM - Kelley-Hoskins Nathan

Status:	Closed	**Start date:**	06/26/2019
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.7.0		

Description

If you give GTime.utc(string) a utc string, it sets itself to that value. However, if the seconds in the string are 59.5 or higher, I think the string gets rounded up, and the fractional seconds get zeroed out.

Example script:

```
#!/usr/bin/env python
import gammalib, random

random.seed(12)
for i in range(30) :
  r = 59.0 + random.random()
  s = '2019-07-03T08:20:%.6f' % r
  #print(s)
  g = gammalib.GTime()
  g.utc(s)
  print('%.3f'%r,'|',s, '|', g.utc())
```

Output:

```
seconds| input utc() string         | output utc() string
59.475 | 2019-07-03T08:20:59.474571 | 2019-07-03T08:20:59.474571
59.657 | 2019-07-03T08:20:59.657473 | 2019-07-03T08:21:0.000000   <- ack
59.666 | 2019-07-03T08:20:59.666410 | 2019-07-03T08:21:0.000000   <- ack
59.143 | 2019-07-03T08:20:59.142600 | 2019-07-03T08:20:59.142600
59.011 | 2019-07-03T08:20:59.010860 | 2019-07-03T08:20:59.010860
59.375 | 2019-07-03T08:20:59.374754 | 2019-07-03T08:20:59.374754
59.274 | 2019-07-03T08:20:59.274048 | 2019-07-03T08:20:59.274048
59.810 | 2019-07-03T08:20:59.810348 | 2019-07-03T08:21:0.000000   <- ack
59.691 | 2019-07-03T08:20:59.690593 | 2019-07-03T08:21:0.000000   <- ack
59.601 | 2019-07-03T08:20:59.601457 | 2019-07-03T08:21:0.000000   <- ack
59.558 | 2019-07-03T08:20:59.558190 | 2019-07-03T08:21:0.000000   <- ack
59.661 | 2019-07-03T08:20:59.661321 | 2019-07-03T08:21:0.000000   <- ack
59.145 | 2019-07-03T08:20:59.145303 | 2019-07-03T08:20:59.145303
59.440 | 2019-07-03T08:20:59.440055 | 2019-07-03T08:20:59.440055
59.162 | 2019-07-03T08:20:59.162267 | 2019-07-03T08:20:59.162267
59.906 | 2019-07-03T08:20:59.905973 | 2019-07-03T08:21:0.000000   <- ack
59.059 | 2019-07-03T08:20:59.058824 | 2019-07-03T08:20:59.058824
59.819 | 2019-07-03T08:20:59.818820 | 2019-07-03T08:21:0.000000   <- ack
59.075 | 2019-07-03T08:20:59.074610 | 2019-07-03T08:20:59.074610
59.687 | 2019-07-03T08:20:59.686946 | 2019-07-03T08:21:0.000000   <- ack
59.337 | 2019-07-03T08:20:59.337000 | 2019-07-03T08:20:59.337000
59.405 | 2019-07-03T08:20:59.404614 | 2019-07-03T08:20:59.404614
59.842 | 2019-07-03T08:20:59.842403 | 2019-07-03T08:21:0.000000   <- ack
59.019 | 2019-07-03T08:20:59.018604 | 2019-07-03T08:20:59.018604
59.061 | 2019-07-03T08:20:59.060785 | 2019-07-03T08:20:59.060785
59.915 | 2019-07-03T08:20:59.915034 | 2019-07-03T08:21:0.000000   <- ack
59.509 | 2019-07-03T08:20:59.508926 | 2019-07-03T08:21:0.000000   <- ack
59.091 | 2019-07-03T08:20:59.090978 | 2019-07-03T08:20:59.090978
```

59.987 | 2019-07-03T08:20:59.987134 | 2019-07-03T08:21:0.000000 <- ack
59.947 | 2019-07-03T08:20:59.946713 | 2019-07-03T08:21:0.000000 <- ack

I looked in GTime::utc(string), but nothing jumped out at me. I did notice that 0.5 seconds is added and subtracted to avoid a rounding problem #2177 , but I dont know if thats related.

History

#1 - 06/27/2019 07:10 AM - Knödlseder Jürgen

Isn't this what you would expect for a format that shows no digits after the comma?

#2 - 06/24/2020 09:58 PM - Knödlseder Jürgen

- Assigned To set to Knödlseder Jürgen

- Target version set to 1.7.0

Using the code

```
#!/usr/bin/env python
import gammalib, random

random.seed(12)
for i in range(30) :
  r = 59.0 + random.random()
  s = '2019-07-03T08:20:%.6f' % r
  #print(s)
  g = gammalib.GTime()
  g.utc(s)
  print('%.3f'%r,'|',s, '|', g.utc(3))
```

results in

```
('59.475', '|', '2019-07-03T08:20:59.474571', '|', '2019-07-03T08:20:59.475')
('59.657', '|', '2019-07-03T08:20:59.657473', '|', '2019-07-03T08:21:00.000')
('59.666', '|', '2019-07-03T08:20:59.666410', '|', '2019-07-03T08:21:00.000')
('59.143', '|', '2019-07-03T08:20:59.142600', '|', '2019-07-03T08:20:59.143')
('59.011', '|', '2019-07-03T08:20:59.010860', '|', '2019-07-03T08:20:59.011')
('59.375', '|', '2019-07-03T08:20:59.374754', '|', '2019-07-03T08:20:59.375')
('59.274', '|', '2019-07-03T08:20:59.274048', '|', '2019-07-03T08:20:59.274')
('59.810', '|', '2019-07-03T08:20:59.810348', '|', '2019-07-03T08:21:00.000')
('59.691', '|', '2019-07-03T08:20:59.690593', '|', '2019-07-03T08:21:00.000')
('59.601', '|', '2019-07-03T08:20:59.601457', '|', '2019-07-03T08:21:00.000')
('59.558', '|', '2019-07-03T08:20:59.558190', '|', '2019-07-03T08:21:00.000')
('59.661', '|', '2019-07-03T08:20:59.661321', '|', '2019-07-03T08:21:00.000')
('59.145', '|', '2019-07-03T08:20:59.145303', '|', '2019-07-03T08:20:59.145')
('59.440', '|', '2019-07-03T08:20:59.440055', '|', '2019-07-03T08:20:59.440')
('59.162', '|', '2019-07-03T08:20:59.162267', '|', '2019-07-03T08:20:59.162')
('59.906', '|', '2019-07-03T08:20:59.905973', '|', '2019-07-03T08:21:00.000')
('59.059', '|', '2019-07-03T08:20:59.058824', '|', '2019-07-03T08:20:59.059')
('59.819', '|', '2019-07-03T08:20:59.818820', '|', '2019-07-03T08:21:00.000')
('59.075', '|', '2019-07-03T08:20:59.074610', '|', '2019-07-03T08:20:59.075')
('59.687', '|', '2019-07-03T08:20:59.686946', '|', '2019-07-03T08:21:00.000')
('59.337', '|', '2019-07-03T08:20:59.337000', '|', '2019-07-03T08:20:59.337')
('59.405', '|', '2019-07-03T08:20:59.404614', '|', '2019-07-03T08:20:59.405')
('59.842', '|', '2019-07-03T08:20:59.842403', '|', '2019-07-03T08:21:00.000')
('59.019', '|', '2019-07-03T08:20:59.018604', '|', '2019-07-03T08:20:59.019')
('59.061', '|', '2019-07-03T08:20:59.060785', '|', '2019-07-03T08:20:59.061')
('59.915', '|', '2019-07-03T08:20:59.915034', '|', '2019-07-03T08:21:00.000')
('59.509', '|', '2019-07-03T08:20:59.508926', '|', '2019-07-03T08:21:00.000')
('59.091', '|', '2019-07-03T08:20:59.090978', '|', '2019-07-03T08:20:59.091')
('59.987', '|', '2019-07-03T08:20:59.987134', '|', '2019-07-03T08:21:00.000')
('59.947', '|', '2019-07-03T08:20:59.946713', '|', '2019-07-03T08:21:00.000')
```

#3 - 06/24/2020 10:14 PM - Knödlseder Jürgen

I corrected the computation in GTime::utc() by using a precision dependent margin that is added and later subtracted to the second in order to avoid seconds of 60. The output is now

('59.475', '|', '2019-07-03T08:20:59.474571', '|', '2019-07-03T08:20:59.475')
('59.657', '|', '2019-07-03T08:20:59.657473', '|', '2019-07-03T08:20:59.657')
('59.666', '|', '2019-07-03T08:20:59.666410', '|', '2019-07-03T08:20:59.666')
('59.143', '|', '2019-07-03T08:20:59.142600', '|', '2019-07-03T08:20:59.143')
('59.011', '|', '2019-07-03T08:20:59.010860', '|', '2019-07-03T08:20:59.011')
('59.375', '|', '2019-07-03T08:20:59.374754', '|', '2019-07-03T08:20:59.375')
('59.274', '|', '2019-07-03T08:20:59.274048', '|', '2019-07-03T08:20:59.274')
('59.810', '|', '2019-07-03T08:20:59.810348', '|', '2019-07-03T08:20:59.810')
('59.691', '|', '2019-07-03T08:20:59.690593', '|', '2019-07-03T08:20:59.691')
('59.601', '|', '2019-07-03T08:20:59.601457', '|', '2019-07-03T08:20:59.601')
('59.558', '|', '2019-07-03T08:20:59.558190', '|', '2019-07-03T08:20:59.558')
('59.661', '|', '2019-07-03T08:20:59.661321', '|', '2019-07-03T08:20:59.661')
('59.145', '|', '2019-07-03T08:20:59.145303', '|', '2019-07-03T08:20:59.145')
('59.440', '|', '2019-07-03T08:20:59.440055', '|', '2019-07-03T08:20:59.440')
('59.162', '|', '2019-07-03T08:20:59.162267', '|', '2019-07-03T08:20:59.162')
('59.906', '|', '2019-07-03T08:20:59.905973', '|', '2019-07-03T08:20:59.906')
('59.059', '|', '2019-07-03T08:20:59.058824', '|', '2019-07-03T08:20:59.059')
('59.819', '|', '2019-07-03T08:20:59.818820', '|', '2019-07-03T08:20:59.819')
('59.075', '|', '2019-07-03T08:20:59.074610', '|', '2019-07-03T08:20:59.075')
('59.687', '|', '2019-07-03T08:20:59.686946', '|', '2019-07-03T08:20:59.687')
('59.337', '|', '2019-07-03T08:20:59.337000', '|', '2019-07-03T08:20:59.337')
('59.405', '|', '2019-07-03T08:20:59.404614', '|', '2019-07-03T08:20:59.405')
('59.842', '|', '2019-07-03T08:20:59.842403', '|', '2019-07-03T08:20:59.842')
('59.019', '|', '2019-07-03T08:20:59.018604', '|', '2019-07-03T08:20:59.019')
('59.061', '|', '2019-07-03T08:20:59.060785', '|', '2019-07-03T08:20:59.061')
('59.915', '|', '2019-07-03T08:20:59.915034', '|', '2019-07-03T08:20:59.915')
('59.509', '|', '2019-07-03T08:20:59.508926', '|', '2019-07-03T08:20:59.509')
('59.091', '|', '2019-07-03T08:20:59.090978', '|', '2019-07-03T08:20:59.091')
('59.987', '|', '2019-07-03T08:20:59.987134', '|', '2019-07-03T08:20:59.987')
('59.947', '|', '2019-07-03T08:20:59.946713', '|', '2019-07-03T08:20:59.947')

#4 - 06/24/2020 10:19 PM - Knödlseder Jürgen

- Status changed from New to Pull request

- % Done changed from 0 to 100

The following code demonstrates that the rounding is done correctly:

```
#!/usr/bin/env python
import gammalib

times = ['2019-07-03T08:20:59.0000',
         '2019-07-03T08:20:59.9000',
         '2019-07-03T08:20:59.9900',
         '2019-07-03T08:20:59.9990',
         '2019-07-03T08:20:59.9999']

g = gammalib.GTime()
for time in times:
    g.utc(time)
    print(time, '|', g.utc(0), g.utc(1), g.utc(2), g.utc(3), g.utc(10))
```

which produces

```
('2019-07-03T08:20:59.0000', '|', '2019-07-03T08:20:59', '2019-07-03T08:20:59.0', '2019-07-03T08:20:59.00', '2019-07-03T08:20:59.000',
 '2019-07-03T08:20:59.0000003111')
('2019-07-03T08:20:59.9000', '|', '2019-07-03T08:21:00', '2019-07-03T08:20:59.9', '2019-07-03T08:20:59.90', '2019-07-03T08:20:59.900',
 '2019-07-03T08:20:59.8999998299')
('2019-07-03T08:20:59.9900', '|', '2019-07-03T08:21:00', '2019-07-03T08:21:00.0', '2019-07-03T08:20:59.99', '2019-07-03T08:20:59.990',
 '2019-07-03T08:20:59.9900000962')
('2019-07-03T08:20:59.9990', '|', '2019-07-03T08:21:00', '2019-07-03T08:21:00.0', '2019-07-03T08:21:00.00', '2019-07-03T08:20:59.999',
 '2019-07-03T08:20:59.9989997456')
('2019-07-03T08:20:59.9999', '|', '2019-07-03T08:21:00', '2019-07-03T08:21:00.0', '2019-07-03T08:21:00.00', '2019-07-03T08:21:00.000',
 '2019-07-03T08:20:59.9998999620')
```

#5 - 06/25/2020 07:50 AM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into devel.