# GammaLib - Bug #1324

# GApplicationPars only stores reals to 6 digit precision

09/25/2014 12:20 PM - Kelley-Hoskins Nathan

Status:	Closed	Start date:	09/25/2014
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	00-09-00		
Description         Ran into a problem where saving a parameter in ctselect was not working. If you give a parameter a number with more that 6 digits of precision, they get rounded to 6. You can reproduce it with the following commands:         \$ python         import ctools         sel = ctools.select()         sel['tmin'].real(1234567.89)			
print sel['tmin'].real()			

1234570.0

sel['tmin'].real(3.546546465e9) print sel['tmin'].real()

3546550000.0

I was hoping to set tmin and tmax to something larger (i.e. some 9- or 10-digit-precision time in seconds in veritas's MET time), but this prevents me.

#### History

#1 - 09/25/2014 12:24 PM - Kelley-Hoskins Nathan

- Description updated

Example should have been formatted like this instead:

\$ python
>>> import ctools
>>> sel = ctools.select()
>>> sel['tmin'].real(1234567.89)
>>> print sel['tmin'].real()
1234570.0
>>> sel['tmin'].real(3.546546465e9)
>>> print sel['tmin'].real()
3546550000.0

#### #2 - 09/25/2014 02:05 PM - Mayer Michael

- % Done changed from 0 to 20

I dived a bit into the GApplicationPar-class. As I understand, the problem occurs in the function

std::string gammalib::str(const double &value)

in GTools.C.

The problem, however, has nothing to do with the gammalib library. It is more likely a c++ feature when casting a double into a string. The same happens if you have a double and try to print it on the screen with std::cout;

Luckily, gammalib::str() can take the precision as an argument. Therefore, I guess for the time cut parameters (maybe for any parameter?) we could change line 423 of GApplicationPar.C to something like

std::string value\_string = gammalib::str(value,5);

or any other value. This should quickly solve this issue. In the long run, we might think about handling the time cuts with a different precision than e.g. energy or coordinate cuts.

#### #3 - 09/30/2014 10:12 PM - Knödlseder Jürgen

I changed the code in GApplicationPar::real(double) as follows:

```
// Set value string at highest precision
std::string value_string = gammalib::str(value, 15);
// Strip trailing zeros
std::string::size_type start = 0;
std::string::size_type stop = value_string.find_last_not_of("0");
if (stop != std::string::npos) {
    if (start <= stop) {
        std::string tmp = value_string.substr(start, stop-start+1);
        value_string = tmp;
    }
}</pre>
```

The 15 is the maximum allowed precision for a double precision variable. The remaining code strips any trailing zeros from the string so that we don't store always many zero digits for a real variable in the string.

I merged the code into the devel branch. Please let me know if this gives reasonable results on your side.

## #4 - 09/30/2014 10:12 PM - Knödlseder Jürgen

- Status changed from New to Feedback
- % Done changed from 20 to 80

### #5 - 09/30/2014 10:12 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

# #6 - 09/30/2014 10:13 PM - Knödlseder Jürgen

- Assigned To set to Mayer Michael
- Target version set to 00-09-00

# #7 - 10/02/2014 01:24 PM - Kelley-Hoskins Nathan

- Status changed from Feedback to Resolved
- % Done changed from 80 to 100

Pulled devel and tested it, real parameters are stored to 15 digits now. Thanks.

#### #8 - 10/03/2014 03:54 PM - Knödlseder Jürgen

- Status changed from Resolved to Closed