

## GammaLib - Bug #1509

### python unicode strings raise exceptions for c++ methods that require std::string arguments

07/09/2015 12:36 PM - Kelley-Hoskins Nathan

<b>Status:</b>	New	<b>Start date:</b>	07/09/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>Using a python string seems to be ok, but a python unicode string doesn't seem to get converted to a std::string object, see below examples:</p>			
<pre>&gt;&gt;&gt; import gammalib &gt;&gt;&gt; g = gammalib.GModelSky() &gt;&gt;&gt; g.name( 'blah' ) # works fine &gt;&gt;&gt; g.name( u'blah' ) # raises exception Traceback (most recent call last):   File "&lt;stdin&gt;", line 1, in &lt;module&gt;   File "/Users/nkelhos/Software/gammalib-git/lib/python2.7/site-packages/gammalib/model.py", line 413, in name     return _model.GModel_name(self, *args) NotImplementedError: Wrong number or type of arguments for overloaded function 'GModel_name'. Possible C/C++ prototypes are:   GModel::name() const   GModel::name(std::string const &amp;)</pre>			
<pre>&gt;&gt;&gt; import gammalib &gt;&gt;&gt; g = gammalib.GCTAObservation('VERITAS') &gt;&gt;&gt; g.load( 'ctoolsdata/VR75960.chunk2.fits' ) # works fine &gt;&gt;&gt; g.load( u'ctoolsdata/VR75960.chunk2.fits' ) # raises exception Traceback (most recent call last):   File "&lt;stdin&gt;", line 1, in &lt;module&gt;   File "/afs/ifs.de/user/n/nkelhos/scratch/gammalib/gammalib-git/lib64/python2.6/site-packages/gammalib/cta.py", line 231, in load     return _cta.GCTAObservation_load(self, *args) NotImplementedError: Wrong number of arguments for overloaded function 'GCTAObservation_load'. Possible C/C++ prototypes are:   load(GCTAObservation *,std::string const &amp;)   load(GCTAObservation *,std::string const &amp;,std::string const &amp;,std::string const &amp;,std::string const &amp;)</pre>			
<p>A workaround is to encode the unicode string back to ascii, via:</p>			
<pre>n = u'mystring' nascii = n.encode('ascii')</pre>			
<p>This may or may not be a bug that requires a fix, but I thought I'd document the behavior anyway.</p>			