

ctools - Change request #564

ctbin should read input FITS file data after the user enters all parameters

10/13/2012 06:42 PM - Deil Christoph

Status:	Rejected	Start date:	10/13/2012
Priority:	High	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	1.0.0		
Description			
I ran ctbin on a 3.7 GB events.fits input file generated with ctobssim. It prompts for the filename			
Input event list or observation definition file [mucho_events.fits]			
and when I hit enter it reads the 3.7 GB from disk (and makes me wait for a long time) before prompting for the next parameter:			
Start value for first energy bin in TeV [0.1]			
and all the other parameters.			
I think all tools (I didn't check the others) should first read all parameters, then process the data while I go for a coffee. Would there be a problem changing ctbin to work that way?			
The "problem" is that ctbin::get_parameters calls obs.load_unbinned(m_evfile); before getting the remaining parameters:			
<pre>void ctbin::get_parameters(void) { // If there are no observations in container then add a single CTA // observation using the parameters from the parameter file if (m_obs.size() == 0) { // Get name of CTA events file m_evfile = (*this)["evfile"].filename(); // Allocate CTA observation GCTAObservation obs; // Try first to open as FITS file try { // Load event list in CTA observation obs.load_unbinned(m_evfile); // Append CTA observation to container m_obs.append(obs); // Signal that no XML file should be used for storage m_use_xml = false; } // ... otherwise try to open as XML file catch (GException::fits_open_error &e) { // Load observations from XML file m_obs.load(m_evfile);</pre>			

```

// Signal that XML file should be used for storage
m_use_xml = true;

}

// Use the xref and yref parameters for binning (otherwise the
// pointing direction(s) is/are used)
//m_xref = (*this)["xref"].real();
//m_yref = (*this)["yref"].real();

} // endif: there was no observation in the container

// Get remaining parameters
m_emin = (*this)["emin"].real();
m_emax = (*this)["emax"].real();
m_enumbins = (*this)["enumbins"].integer();
m_proj = (*this)["proj"].string();
m_coordsys = (*this)["coordsys"].string();
m_xref = (*this)["xref"].real();
m_yref = (*this)["yref"].real();
m_binsz = (*this)["binsz"].real();
m_nxpix = (*this)["nxpix"].integer();
m_nypix = (*this)["nypix"].integer();

// Optionally read ahead parameters so that they get correctly
// dumped into the log file
if (m_read_ahead) {
    m_outfile = (*this)["outfile"].filename();
    m_prefix = (*this)["prefix"].string();
}

// Return
return;
}

```

Related issues:

Related to ctools - Change request # 1350: Separate parameter reading from fi...	Rejected	10/30/2014
Related to GammaLib - Action # 1349: GApplication::log_parameters should only...	Closed	10/30/2014

History

#1 - 10/30/2014 12:30 PM - Knödlseeder Jürgen

- Priority changed from Normal to High
- Target version set to 1.0.0

We need to make sure that this is fixed for release version 1.0.0.

#2 - 10/30/2014 02:05 PM - Mayer Michael

For such a large file as described above it makes sense to do the reading on start of the run()-method. In my opinion, we could make the following change:

1. get_parameters() just checks whether the file is existent (or sanity of other parameters)
2. run() actually loads the files before executing the analysis step

We might also think about including read and load methods to the ctools base-class. read() could read the parameters and checks for sanity, while load() actually reads the input FITS files.

The only concern I have is that an error, which occurs on reading a file, might raise rather late, when Christoph already went for coffee smile.png

#3 - 10/30/2014 02:40 PM - Knödseder Jürgen

It's in fact not easy to make a complete separation, at least for the moment.

Reading of user parameters is in general conditional to what the ctool actually knows already. For example, if the ctool has already an observation container, it won't read observation data. Same for a model definition file. In some cases, a ctool needs to open a file (or check at least its existence) to decide what to do next. So one needs to carefully check what can be done.

In the long run I would agree that we should have a section that reads the parameters and another that loads information, but we maybe should wait with introducing that until we understand better how to deal in a general way with the problem. I created an issue for that (#1350).

As a side effect, there is an annoying issue with the parameters that are dumped in the log file: they may be misleading as they are eventually not used, but the user does not know this. I create an action #1349 for that, but that action is only efficient until a clear separation is made (issue #1350).

So let's keep this issue for fixing the specific ctbin problem.

#4 - 06/30/2015 11:46 AM - Knödseder Jürgen

- *Status changed from New to Rejected*

Decided that it's not so important for end users. Would be nice, but a pain to implement.