GammaLib - Feature #3662

Add stamp() methods to GApplication class to stamp FITS HDUs, objects and files with provenance information

05/08/2021 11:16 AM - Knödlseder Jürgen

Status:	Closed		**Start date:**	05/08/2021
Priority:	Normal		**Due date:**	
Assigned To:	Knödlseder Jürgen		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:	2.0.0			

Description

Methods should be added to the GApplication class to stamp FITS HDUs with provenance information. Minimal provenance information is the name and version of the application, but also application parameters can be written into the FITS header.

The following methods should be added:

void stamp(GFitsHDU& hdu) const;
void stamp(GFits& fits) const;
void stamp(const GFilename& filename) const;

The first method stamps a single HDU, the second method stamps all HDUs in a FITS object, and the last method stamps all HDUs in a FITS file.

History

#1 - 05/08/2021 10:54 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 10

I added stamp() methods to the GApplication class that either stamp a FITS header, a FITS object or a FITS file with provenance information. So far only the CREATOR keyword is written using format recommended by R7 of https://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/ofwg_recomm.html:

name vx.y.z

where name is the application name and x.y.z is the version number.

#2 - 05/09/2021 09:41 AM - Knödlseder Jürgen

- % Done changed from 10 to 20

I merged the code into devel.

Next step is to modify the ctools so that all written FITS files are stamped.

#3 - 05/10/2021 10:08 AM - Tibaldo Luigi

I think at least the 'infile' and 'inmodel' parameters would be very useful to have in the output header

#4 - 05/10/2021 11:55 AM - Knödlseder Jürgen

user#266 wrote:

> I think at least the 'infile' and 'inmodel' parameters would be very useful to have in the output header

Fully agree. Maybe we just should add all parameters. Note however that input information can also directly provided if the scripts are used in Python (e.g. passing of an observation container) and it's not fully clear to me how to track this information. Maybe we should wait here also for some guidelines from CTAO on how to track provenance information. I guess some keyword conventions are needed here.

#5 - 05/20/2021 03:43 PM - Knödlseder Jürgen

- % Done changed from 20 to 30

I added stamping to the following ctools:

- ctbin
- ctbkgcube
- ctcubemask
- ctedispcube
- ctexpcube
- ctfindvar
- ctmapcube
- ctmodel
- ctobssim
- ctpsfcube
- ctskymap
- cttsmap
- ctool::save_event_list

#6 - 05/20/2021 04:00 PM - Knödlseder Jürgen

I added stamping to the following cscripts:

- csebins
- cslightcrv
- csmodelsois
- csobs2caldb
- csphagen
- csphasecrv
- csresmap
- csresspec
- csroot2caldb
- csscs
- csspec
- cststmapmerge
- csviscube

#7 - 05/20/2021 05:34 PM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 30 to 100

Code was integrated into devel, close issue now.