ctools - Bug #3066

Remove second header in log files of some cscripts

11/15/2019 10:08 AM - Knödlseder Jürgen

Status:	Closed	**Start date:**	11/15/2019
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.7.0		

Description

When running some cscripts from the command line the header appears twice in the log file:

```
2019-11-15T08:55:49: ********************************************************************************
2019-11-15T08:55:49: *                          csviscube                           *
2019-11-15T08:55:49: * ---------------------------------------------------------------- *
2019-11-15T08:55:49: * Version: 1.7.0.dev                                             *
2019-11-15T08:55:49: ********************************************************************************
2019-11-15T08:55:49: ********************************************************************************
2019-11-15T08:55:49: *                          csviscube                           *
2019-11-15T08:55:49: * ---------------------------------------------------------------- *
2019-11-15T08:55:49: * Version: 1.7.0.dev                                             *
2019-11-15T08:55:49: ********************************************************************************
2019-11-15T08:55:51: +=============+
2019-11-15T08:55:51: | Parameters |
2019-11-15T08:55:51: +=============+
```

This feature does not appear when the scripts are run from Python. Here the scripts that seem to be concerned:

- cscaldb
- csfindobs
- csiactcopy
- csiactdata
- csiactobs
- csmodelinfo
- csmodelmerge
- csmodelsois
- csobsdef
- csroot2caldb
- cssrcdetect
- cstsmapmerge
- csviscube
- csworkflow

The common feature of these scripts is that they all derive from ctools.cscript. I only sampled the other scripts, but they seem to derive from derived classes.

History

#1 - 11/15/2019 11:11 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 10

The difference came from

```
# Execute the script
def _execute(self):
```

```
    self.logFileOpen()    # This one was not in ctool.i
    self._read_ahead(True)
    self.run()
    self.save()
cscript.execute = _execute
%}
```

between ctool.i, ctobservation.i and ctlikelihood.i, where the opening of the logfile was missing in ctool.i.

Note that this implies that the logfile is written when the execute method is called from Python!

#2 - 11/15/2019 11:21 AM - Knödlseder Jürgen

I turned out that the line

```
self.logFileOpen()
```

was removed in #2182 in the ctool.i file but not the other files. The side effect of this is that the header is written twice, since the single header was in fact from a re-opening of the logfile that erases everything written earlier. This is illustrated below:

```
$ cstsdist
Enter ctool::ctool                  # First opening of log file
Enter GLog::open
Exit GLog::open
Enter GApplication::log_header      # First writing of header
Exit GApplication::log_header
Exit ctool::ctool
Enter ctobservation::ctobservation
Exit ctobservation::ctobservation
Enter GApplication::log_header      # Second writing of header
Exit GApplication::log_header
Enter GLog::open                    # Reopening of log file
Exit GLog::open
Enter GApplication::log_header      # First writing of header in new log file
Exit GApplication::log_header
```

Note that I added debugging print statements to GammaLib to better understand what's going on.

#3 - 11/15/2019 11:27 AM - Knödlseder Jürgen

- % Done changed from 10 to 50

After removing the call to self.logFileOpen() also in ctobservation.i and ctlikelihood.i, here what you get when calling a script that derives from ctobservation.i in Python:

```
>>> import cscripts
>>> a=cscripts.cstsdist()
Enter GApplication::log_header
Exit GApplication::log_header
>>> a.execute()
```

No log file is created.

And here what happens when calling the script from the command line:

```
$ cstsdist
Enter ctool::ctool
Enter GLog::open
Exit GLog::open
Enter GApplication::log_header
Exit GApplication::log_header
Exit ctool::ctool
Enter ctobservation::ctobservation
Exit ctobservation::ctobservation
Enter GApplication::log_header
Exit GApplication::log_header
```

Now a log file is created with two headers.

#4 - 11/15/2019 11:34 AM - Knödlseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 50 to 100

I removed the self._log_header() call in the init methods of the scripts:

```
    # Set logger properties
    #self._log_header()
    self._log.date(True)
```

This then leads to

```
$ cstsdist
Enter ctool::ctool
Enter GLog::open
Exit GLog::open
Enter GApplication::log_header
Exit GApplication::log_header
Exit ctool::ctool
Enter ctobservation::ctobservation
Exit ctobservation::ctobservation
```

when the script is called from the command line and only one header is present.

And here is what happens when calling now execute from Python after opening the log file:

```
>>> import cscripts
>>> a=cscripts.cstsdist()
>>> a.logFileOpen()
Enter GLog::open
Exit GLog::open
Enter GApplication::log_header
Exit GApplication::log_header
>>> a.execute()
```

A log file with a single header is present.

This is the desired behavior.

#5 - 11/15/2019 02:14 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into devel.