odfcheck

January 12, 2017

Abstract

Validates ODFs

1 Instruments/Modes

Instrument	Mode

2 Use

pipeline processing	no
interactive analysis	no

3 Description

odfcheck checks that an ODF meets the quality requirements as described in **XMM-SOC-USR-URD-0003 (Guainazzi et al)**. **odfcheck** works by performing a number of tests. Each test can be individually switched off.

Before running **odfcheck** the environment variables `SAS_ODF` and `SAS_CCFPATH` should be set to point to the ODF to be checked and the location where the CCF constituents can be found.

3.1 Preliminaries

odfcheck first runs **cifbuild** and **odfingest** on the ODF. This creates a calibration index file and a summary file for further use. This step can be skipped (see `skippreliminaries`) in which case the variable `SAS_CCF` should be set to point to an existing CIF, and the variable `SAS_ODF` should point to the summary file generated by **odfingest**.

3.2 Attitude check

This check processes the Attitude History File in order to determine the stability of the attitude reconstruction. **odfcheck** estimates the stability of the attitude reconstruction by calculating the r.m.s. of the parameters `DIFFVRA` and `DIFFVDEC` in the AHF, during the stable pointing period (after the completion of the closed loop slew where `VALTIME>PTTIME`).

If either of the parameters exceeds a user supplied threshold, a warning is issued.

3.3 Time correlation check

This check makes use of the **oal** to determine how good the fit to the time correlation data is. If the fraction of data points that are discarded in the fit is larger than a user-specified threshold, a warning is issued.

3.4 Telemetry check

This steps performs a statictical estimate of the possible telemetry loss. It does so by looking at the histogram of the time difference between consecutive time packets found in the time correlation file. All differences larger than a user-specified threshold are listed, and from these an estimated telemetry loss fraction is calculated. If this fraction is larger than the value given by the user, a warning is issued.

3.5 Counting mode check

This check identifies the time intervals when any of the EPIC instruments is in counting mode. This is done by looking at the counting cycle report data in the ODF. The following anomalies are detected and lead to a warning:

- A counting mode interval appears from the data to have a negative duration.
- A counting mode interval appears to have lasted longer than the exposure it belongs to.

These anomalies usually point to problems with the creation of the ODF.

3.6 OM priority field acquisition check

The consistency and validity of the data contained in the priority field acquisition data files is checked for. The algorithm as specified in the DOCUMENT is used to perform this check. Warnings are issued if any of the conditions are not met.

3.7 Housekeeping GTI check

For each instrument **odfcheck** generates a GTI based on the HKPARMINT CCF constituents.

3.8 EMOS temperature check

This checks identifies any EMOS exposures when the instrument temperatue was lower than a user-specified threshold. A warning is generated for each occurrence.

3.9 EPN energy check

The consistency between the energy values reported in the data file for EPN and the housekeeping values descibing the hardware setting of the instrument is checked for. A warning is issued if any events are found to have energy below the hardware threshold.

3.10 EPN minimum energy check

The energy values reported in data files for the PN modes imaging, timing, and burst, are checked against a user-specified value. A warning is issued if any entry is below the threshold.

3.11 SAS compliance check

odfcheck verifies that every FITS data file in the ODF can be read by the **dal**. This is done by opening each datasets in high memory mode, thereby forcing the whole of the data to be read.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
skippreliminaries	no	bool	no	

If set **odfcheck** will assume that the user has already run **cifbuild** and **odfingest**, and that the relevant environment variables have been set. Otherwise **cifbuild** and **odfingest** will be run. The resulting CIF will have a name like RRRR_XXXXXXXXXX.cif, where RRRR_XXXXXXXXXX is the revolution/observation identifier for the current ODF.

checkattitude	no	bool	yes	

If set **odfcheck** will run the attitude diagnostics on the current ODF.

allowedbadattitudefraction	no	real	65	[0,100]

When running the attitude diagnostics **odfcheck** will issue a warning if the attitude is more than `allowedbadattitudefraction`% of the time more than `nrms` *r.m.s.* away from the median.

nrms	no	real	3	> 0

If the attitude diagnostic is being run, then the fraction of the time that the attitude differs more than `nrms` from the median pointing will be calculated.

allowedtimecorrelationfraction	no	real	65	[0,100]

When running the time correlation diagnostic **odfcheck** will issue a warning if more than `allowedtimecorrelationfraction`% of the available data points need to be discarded in order to reach a good fit.

checktimecorrelation	no	bool	yes	

If set **odfcheck** will run the time correlation diagnostics on the current ODF.

tcpinterval	no	real	35	> 0

If the time correlation diagnostic is being run, time packets differing more than `tcpinterval` seconds from each other will be considered in estimating the statistical loss of telemetry.

checktelemetry	no	bool	yes	

If set **odfcheck** will run the telemetry loss diagnostics on the current ODF.

allowedtelemetrylossfraction	no	real	5	[0,100]

When running the telemetry loss diagnostic **odfcheck** will issue a warning if the estimated telemetry loss is larger than `allowedtelemetrylossfraction`%.

checkcountingmode	no	bool	yes	

If set **odfcheck** will run the EPIC counting mode diagnostics on the current ODF.

checkminimumpnenergy	no	bool	yes	

If set **odfcheck** will run the PN minimum energy diagnostics on the current ODF.

checkompriorityfieldacquisition	no	bool	yes	

If set **odfcheck** will run the OM priority field acquisition diagnostics on the current ODF.

checkhkgti	no	bool	yes	

If set **odfcheck** will run the housekeeping GTI diagnostics on the current ODF.

checkmostemperature	no	bool	yes	

If set **odfcheck** will check if the EMOS cameras mean temperature during the science exposures was above the thresholds given with `m1temperature` and `m2temperature`.

m1temperature	no	real	33	

If the corresponfing diagnostics is activated **odfcheck** will use this value as the minimum mean temperature for the EMOS1 camera.

m2temperature	no	real	30	

If the corresponfing diagnostics is activated **odfcheck** will use this value as the minimum mean temperature for the EMOS2 camera.

checkpnenergy	no	bool	yes	

If set **odfcheck** will check if the energy of every event in the PN science files is above a threshold. The latter is calculated from the housekeeping.

checksascompliance	no	bool	yes	

If set **odfcheck** will check that every FITS file in the ODF can be read with the SAS.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

EnvironmentNotConfigured *(error)*
The variable `SAS_CCFPATH` is not set.

SubTaskFailure *(error)*
One of the subtasks that **cifbuild** has invoked failed, and **odfcheck** cannot recover from this failure.

UnknownError *(error)*
An error was detected but it could not be determined its origin.

NoMasterCalindexSet *(warning)*
odfcheck tried to run **cifbuild** via a Master Calibration Index File, but no suitable one was found.
corrective action: Run **cifbuild** without the MIF

AttitudeDiagnosticsFailed *(warning)*
odfcheck detected an error while running the attitude diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

LargeAttitudeExcursions *(warning)*
odfcheck has detected that the attitude is more than allowedbadattitudefraction% of the time more than `nrms` *r.m.s.* away from the median.
corrective action: Continue.

TimeCorrelationDiagnosticsFailed *(warning)*
odfcheck detected an error while running the time correlation diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

LargeTimeCorrelationLoss *(warning)*
More than allowedtimecorrelationfraction% of the available data points had to be discarded in order to reach a good fit of the time correlation data.
corrective action: Continue.

TelemetryLossDiagnosticsFailed *(warning)*
odfcheck detected an error while running the telemetry loss diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

LargeTelemetryLoss *(warning)*
The estimated telemetry loss was larger than allowedtelemetrylossfraction%.
corrective action: Continue.

EpicCountingModeDiagnosticsFailed *(warning)*
odfcheck detected an error while running the EPIC counting mode diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

EpicCountingModeTimeAnomaly *(warning)*
odfcheck detected an inconsistency in the time records associated with the CCX data files. Two possible conditions are detected: the start and end times of the interval when the instrument is in counting mode are not in the right order, *i.e.* the start is later than the end; the duration of the interval when the instrument is in counting mode is longer than the exposure duration. Details are given in the message attached to the warning.
corrective action: Continue with the current test.

PnMinimumEnergyDiagnosticsFailed *(warning)*
odfcheck detected an error while running the PN minimum energy diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

UnknownError *(warning)*
An error was detected but it could not be determined its origin.
corrective action: Move on to the next set of tests.

OMPAXIterations *(warning)*
odfcheck detected that more than 4 iterations were reported for the OM priority field acquisition.
corrective action: Continue with the current test.

OMPAXGuideStars *(warning)*
odfcheck detected an inconsistency in the number of uploaded and found guide stars. Details are reported.
corrective action: Continue with the current test.

OMPAXDiagnosticsFailed *(warning)*
odfcheck detected an error while running the OM priority field acquisition diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

HousekeepingDiagnosticsFailed *(warning)*
odfcheck detected an error while running the housekeeping diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

HkgtigenFailed *(warning)*
odfcheck detected an error while running **hkgtigen**. The exact error is reported.
corrective action: Continue with the current test.

MosTemperatureDiagnosticsFailed *(warning)*
odfcheck found that one EMOS exposure had a mean temperature lower than the threshold. The exposure number is reported.
corrective action: Continue with the current test.

PnEnergyDiagnosticsFailed *(warning)*
odfcheck found one or more PN science files with an incorrect energy value. The exposure number is reported.
corrective action: Continue with the current test.

SasComplianceDiagnosticsFailed *(warning)*
odfcheck detected an error while running the SAS compliance diagnostics. The exact error is reported.
corrective action: Move on to the next set of tests.

6 Input Files

1.

7 Output Files

1.

8 Algorithm

9 Comments

•

References