# tmcheck

January 12, 2017

**Abstract**

This task reads Raw TM files and performs the same checks carried out by stspproc tasks. Depending on the SAS verbosity, NCTRS frames statistics and packet dump statistics will be dumped.

## 1 Description

This task is intended for reading one single raw TM data files (or a set of them, or a subset of one), and dump basic information about its suitability for stsp processing, or going further, about the amount of valid information which may contain when compared with other files.

## 2 Use

tmcheck –StartTime –EndTime –useS2K –strict –cachedir –rawfile

This simple call may be wrapped by a dedicated caller script, which may automatise the starttime and endtime when a single raw file is entered.

Before calling the executable, - Set the delays.fit locatable for it. - Define the TDLYFILENAME env variable. - Set the SAS_VERBOSITY as needed. - Be sure the raw files are located into the cache dir.

| | |
|---|---|
| pipeline processing | no |
| interactive analysis | no |
| Raw TM analysis | yes |

## 3 Description

tmcheck will open the required raw TM datafiles, and for those DU (NCTRS frames) with the time stamp within the entered time range,
- Datafiles info will be displayed (verbosity 4, sparse)
- DU and Frames statistics will be displayed (verbosity 5, verbose)
- Packet statistics will be displayed (verbosity 6, noisy).
- Packet dumps will be generated (verbosity 7).

Note that current implementation will only use packets belonging to VC0.

# 4 Parameters

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

| Parameter | Mand | Type | Default | Constraints |
|---|---|---|---|---|
| **StartTime** | yes | string | none | YYYY-MM-DDTHH:MM:SS.SS |

Start time of range for processing.

| | | | | |
|---|---|---|---|---|
| **EndTime** | yes | string | none | YYYY-MM-DDTHH:MM:SS.SS |

End time of range for processing.

| | | | | |
|---|---|---|---|---|
| **useS2K** | no | bool | no | none |

Use S2k data files

| | | | | |
|---|---|---|---|---|
| **strict** | no | bool | yes | none |

Abort processing when DU sizes errors are found

| | | | | |
|---|---|---|---|---|
| **cachedir** | no | string | /home/tcxgen/tm/rawtm/ | none |

TM Cache directory

| | | | | |
|---|---|---|---|---|
| **rawfile** | no | string | none | none |

If set, Force to read just this rawfile

# 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.

The normal run will produce some warnings and even errors, which will not prevent the report from being created.

**No data File** *(error)*
Check the entered timespan and that the required Raw TM f iles are located in the cache dir directory.

**OBT reset** *(warning)*
None.
*corrective action:* an OBT reset is detected because the counter does not increase as expected. It may happen that a true OBT reset is present or any Raw TM data rejection process led to the apparent OBT reset.

# 6 Input Files

1. delays.fit, which contain the ground station delays, both the XMCS delays and the Actual (real) delays. Not needed for the processing itself.
2. raw tm data files, as needed.

# 7 Output Files

1. Stsdout, with verbosity details.

# 8 Algorithm

See algorithm of stspproc task for additional information. Basically,

1. - A DU Extractor is generated, with defined start-end boundaries.
2. - A STSP Processor is generated, with defined start-end boundaries.
3. - A NCTRS Processor is generated, hanging from STSP Processor, specialised in VC0 frames.
4. - A Packet Processor is generated, hanging from the STSP Processor.

# 9 Comments

•

# References