gtibuild

April 16, 2023

Abstract

Constructs a GTI table from a plain text description file.

1 Instruments/Modes

Instrument	Mode

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

gtibuild constructs an OGIP-compliant table of Good Time-Intervals (GTI) from a plain text description file. A complex fragmentation of the time-line may be described in terms of good and bad intervals, which may be simple or periodic. The description elements are combined as the union of all the good intervals intersected with the complement of the union of all the bad intervals. In other words, a given point in time is good if it is covered by any good intervals and not also covered by any bad intervals. If no good intervals are specified, a single good interval covering all time is assumed; hence all times not covered by any bad intervals are considered good. The relative order of elements in the description file is always irrelevant. (See Section 6 for a description of the syntax of the description file.)

A periodic interval is a good or bad simple interval repeated through a whole number of periods. Because a periodic interval must be translated into a sequence of explicit simple intervals, an open-ended periodic interval would take infinite space to represent in the output GTI table. Therefore the number of periods must be finite. The syntax for describing a periodic interval is intended to be convenient for the purpose of analyzing two-phase phenomena in the data. The starting and ending times are given, as with simple intervals, but the ending time serves only to specify the final period, which may extend beyond that point. Each period is broken into two segments, one good and the other bad. This may be confusing because it sounds like a periodic interval describes both good and bad intervals at the same time; it does not. A periodic good interval contains none of the bad segments that are implicit in its definition. Similarly a periodic bad interval contains none of the good segments that are implicit in its definition. The periodic interval description syntax is simply a convenient shorthand for a long list of simple intervals, regularly spaced and all having the same length and type.

The periodic interval description designates whether the first segment of each period is good or bad, and the second segment is then implicitly the opposite. Any periodic interval becomes its own complement simply by switching this designation. Regardless of which segment is the good segment, the description also specifies whether it is the set of good segments or the set of bad segments that comprise the periodic interval; that is, whether it is a periodic good interval or a periodic bad interval. This choice must be made carefully based on an understanding of how all the description elements are combined, as explained above. To avoid unexpected results, it is best not to have simple good intervals along with a periodic good interval in the same description file. Having simple bad intervals together with a periodic bad interval in the same description file does not create the possibility of unexpected good time. This is a foible of human cognition, not an asymmetry in the description semantics. If these last two paragraphs have not made much sense yet, try again after reading Section 6, below.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

file	yes	file	gti.text	

The name of the input description file.

table	no	table	gti.ds	

The name of the output GTI table.

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.

badFileName *(error)*
The input description file could not be opened.

negativeTime *(error)*
Negative starting times are not permitted.

tooManyIntervals *(error)*
A periodic interval with an unreasonably large number of segments. The interval must have a finite upper bound.

badSegment *(error)*
Both phase-segments of a periodic interval must have positive duration.

badNumericValue *(error)*
A non-numeric token was encountered where a number is required.

badOperator *(error)*
A bad symbol in the third column: check description syntax.

unexpectedEOL *(error)*
The line ended unexpectedly: check description syntax.

6 Input Files

The input file is formatted in plain text as an unordered sequence of time-interval descriptions. Each non-empty line describes an interval, which may be simple or periodic, and good or bad. All intervals are bounded by a non-negative start and stop time. Zero as the stop time indicates eternity as the upper bound. The first two columns of the file are start and stop. In the third column, which is optional, is one of six operators: '+' or '-', for simple intervals, or '+g', '+b', '-g', '-b' for periodic intervals. Plus indicates a good interval and minus indicates bad. If not specified, the operator defaults to '+'.

Periodic intervals break each period into two phase-segments, one good and the other bad, both having positive duration. The second character of the operator specifies whether the first segment is the good one or the bad one ('g' or 'b', respectively). The lengths of the first and second segments are the fourth and fifth columns, which are required for periodic intervals. The sum of the two segment lengths is the period, and the stop time is automatically adjusted to the end of whatever period it falls in.

Consistently with the SAS conventions, the time values used to build a good time interval are interpreted as offsets from the XMM-Newton Mission Reference Time (MRT, `1998-01-01T00:00:00.00 TT` or `1997-12-31T23:58:56.816 UTC`). For convenience it is possible to specify a different reference time with the keyword `timeref` followed by a time descriptor of the form `yyyy-mm-ddThh:mm:ss[.s]`, as shown in the examples below. Time values are then interpreted as `timeref + value - MRT`. The time `timeref` is taken to be expressed in UTC. Not using `timeref` is equivalent to `timeref 1997-12-31T23:58:56.816`.

Comments may be placed on empty lines and at the end of non-empty lines, preceded by '#'. Times are given as floating-point numbers. The following examples illustrates each point of syntax.

```
 100    0          # Good time extends from 100 seconds onward,
  50 1000 -g 40 20 # punctuated by 20 second bad intervals every
                   # 60 seconds beginning at 90 seconds.
 200  250 -        # The instrument had a brief tantrum here.
```

When using a different `timeref`:

```
timeref 2001-12-20T01:02:03
100 0              # Good time extends from timeref + 100s onward
```

7 Output Files

- An OGIP-compliant Good Time-Interval (GTI) table.

8 Algorithm

9 Comments

- *gtiedit* is expected to supersede this task eventually.

References