hkgtigen

August 16, 2005

Abstract

HouseKeeping parameter driven Good-Time-Interval data set GENeration

1 Instruments/Modes

Instrument	Mode
all (including satellite scope)	not applicable

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

3.1 Motivation

The scientific quality of data acquired with any instrument is usually directly related to the values of its key housekeeping (HK) parameters during the data taking period. Events that were detected during periods when one or more of those parameters was out of valid boundaries must therefore not be taken into account in the generation of images, spectra, and rate curves. This is achieved by filtering out the "bad" events with one or more Good-Time-Interval (GTI) data sets before the product generation takes place. An event is considered to be valid if its arrival time is within one of the GTIs in the set. For more details on this issue, please consult the descriptions of task evselect (see documentation of evselect) and package selectlib (see documentation of selectlib).

hkgtigen generates for a specified instrument a *single* GTI data set from a list of given HK parameter validity intervals. These are read from the Current Calibration File (CCF) constituent *HKParmInt* through the Calibration Access Layer (CAL) and can optionally be augmented by user-specified validity ranges. The created GTI set is applicable to all science data from that instrument in one Observation Data File (ODF). **hkgtigen** makes use of the generic task **tabgtigen** for the actual generation of the GTI data set.

3.2 Detailed Description

For each instrument standard HK parameter validity ranges are stored in the corresponding CCF constituent *HKParmInt* [2]. The ranges are expressed in terms of boolean selection expressions as defined by the selectlib (see documentation of selectlib) package. Each expression is indexed by a unique keyword which is the HK parameter name to which the expression applies. It can contain one or more occurrences of the character '@' which must be replaced with the parameter name to form a complete syntactically correct selection expression. As an example, the *HKParmInt* entry:

```
PYRO_DOOR_RELEASE_MECHANISM_T:   (@ in [120, 273])
```

defines the valid range of the HK parameter *PYRO_DOOR_RELEASE_MECHANISM_T* to be 120 to 273 K. Different entries are combined by **hkgtigen** via the logical *and* operator (`&&`) to form a single boolean expression according to the syntax rules defined in package selectlib (see documentation of selectlib).

The HK parameter names are defined by the names of the corresponding columns in the periodic HK parameter tables of the ODF [1, 3]. Depending on the instrument, there is either one or two types of periodic HK files for each exposure. **hkgtigen** generates a single GTI set which is applicable to the entire observation period, i.e., it must first construct a number of individual GTI sets (one per type of periodic HK file per exposure period) and subsequently merge all sets into one. For the construction of an actual GTI set from a given periodic ODF HK file the task **tabgtigen** is used. The merging of the individual GTI sets is done in task **hkgtigen** itself.

In more detail, when invoked for a particular instrument (see parameter `instrument`), **hkgtigen** performs the following operations to construct the single GTI data set:

1. Retrieve all applicable validity intervals for that instrument from the CCF via the Calibration Access Layer (CAL) (see documentation of cal) and replace all '@'-references with the actual HK parameter names.
2. For each instrument exposure in the given ODF and each type of periodic HK file the following steps are carried out:
   (a) Construct a boolean expression containing validity ranges only for those HK parameters present in the current HK file
   (b) Invoke the task **tabgtigen** with the constructed expression and the name of the current HK file. This generates a new GTI data set.
3. Merge all GTI sets constructed in step 2 into a single GTI data set which is applicable to the entire observation period.

hkgtigen provides a number of parameters to influence this algorithm. For instance, the parameter `parameters` can contain the explicit list of HK parameter names on which the CAL should be queried for validity ranges. It is also possible to augment the list of validity ranges retrieved from the CAL with additional ones. For more information on this and other command line options see below.

3.3 Examples

Please note: In order to fully understand the following examples, please read first the task parameter section:

1. Retrieve the list of all available HK parameter validity ranges for instrument `EPN` from the CAL and construct a GTI data set with name `hkgti_epn.ds`:

```
hkgtigen instrument=epn withgtiset=true gtiset=hkgti_epn.ds
```

2. As 1. but just consider the validity ranges for parameters `P1`, and `P2`:

```
hkgtigen instrument=epn withparameters=true parameters="P1 P2" \
  withgtiset=true gtiset=hkgti_epn.ds
```

3. As 1. but do *not* consider any present validity ranges for parameters *P1*, and *P2*:

```
hkgtigen instrument=epn withparameters=true except=true parameters="P1 P2" \
  withgtiset=true gtiset=hkgti_epn.ds
```

4. As 3.; consider additional validity ranges for two parameters *PS1*, and *PS2*. If the CAL provides ranges for these as well, they get replaced by the values on the command line:

```
hkgtigen instrument=epn withparameters=true except=true parameters="P1 P2" \
  withoverrideparameters=true overrideparameters="PS1 PS2" \
  overrideparametervalues="@ IN [-99, 99] ; @ IN [1234, 5678] || @ IN [10,20]"\
  withgtiset=true gtiset=hkgti_epn.ds
```

3.4 Long and short HK parameter names

Unfortunately the periodic HK parameters in the ODF are stored in table columns with very concise non-descriptive names such as `E605` [3]. For obvious reasons the direct use of these names is very inconvenient and prone to errors. Therefore **hkgtigen** (via a corresponding service in the CAL) implements a name mapping scheme that allows the user to specify the HK parameters (see task parameters `parameters` and `overrideparameters`) with long descriptive names while the ODF access is transparently performed with the short ones. The translation table which drives this process is part of the CCF constituent *HKParmInt*. In order to find out what long parameter names are defined please inspect the contents of *HKParmInt* with the task **calview**. This will also show the list of standard HK parameter validity ranges that **hkgtigen** accesses through the CAL.

3.5 Diagnostic mode

The task possesses a diagnostic mode (see parameter `diagnosticmode`) which facilitates a more detailed analysis of the individual HK parameter validity intervals. In this mode, in addition to the global GTI, each *individual* HK parameter selection expression is used to construct a corresponding GTI table which contributes to the global one. All GTI tables will be generated in a single data set named via the `gtiset` task parameter. The table names are be `STDGTI` (total) and `STDGTI-`*hkname* (individual) respectively. The individual GTI tables contain the corresponding selection expressions in an attribute `EXPR`.

In cases where the total sum of the global GTIs is only a small fraction of the observing time the individual GTIs can be used to pinpoint the most limiting parameter(s). To aid this diagnosis further the individual GTIs together with the global ones can be visualized using task **dsplot** (see parameter `plotgtis`) and an ASCII diagnostic output file is generated (see parameter `diagfile`). The contents is formed by a two-column table - the first column gives the HK parameter name and the second column the fraction of the total observing time that this parameter is out of valid limits. Lines starting with `#` are comments and should be ignored.

3.6 Comments

Please note:

- An invocation of **hkgtigen** for a particular instrument does generate a GTI which covers the duration of the entire observation. This *does not* exclude the possibility to apply this GTI data set to single-exposure event lists. GTIs outside the exposure will simply have no effect in the filtering process.
- The proposed scheme does not exclude the possibility to perform event list filtering with HK parameter GTIs from different instruments. For doing this, please construct the GTI sets for all desired instruments (e.g. `gti_om.ds`, `gti_emos1.ds`, `gti_sc.ds`) and combine them in the selection expression that drives the filtering process in task **evselect**:

```
evselect table=ev.ds expression="TIME in gti(gti_emos1.ds) && \
       TIME in gti(gti_om.ds) && TIME in gti(gti_sc.ds)" ...
```

- Spacecraft HK GTIs can be constructed by invoking **hkgtigen** with `instrument` (Sect. **??**)) set to `xmm`.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
instrument	yes	string	emos1	xmm|emos1|emos2|epn|rgs1|rgs2|om

This determines the instrument for which the GTI data set is to be generated.

withparameters	no	boolean	false	true|false

A boolean switch determining whether the list of to be considered HK parameters shall be explicitly specified with parameter `parameters`. If set to false, all parameters for which entries exists in the CCF file `HKParmInt` are considered in the GTI data set generation.

parameters	no	string list	""	blank separated list of HK parameter names

If `withparameters` is set to `true` only the HK parameters given in the list will be considered in the generation of the final GTI data set. The parameter names must be given in their long (see Sect. 3.4) form.

except	no	boolean	false	true|false

If set to `true` and `withparameters` is set to `true` all HK parameters for which there are entries in the CCF file `HKParmInt` are considered in the GTI set generation except for those explicitly listed in `parameters`.

withoverrideparameters	no	boolean	false	true|false

If set to `true` `overrideparameters` must contain a list of HK parameter names and `overrideparametervalues` a corresponding list of validity ranges for those parameters (see below).

overrideparameters	no	string list	""	blank separated list of HK parameter names

If `withoverrideparameters` is set to `true` the list must contain valid HK parameter names. For each parameter there must be a corresponding value (i.e., validity range expression) specified in parameter `overrideparametervalues`. This value replaces the one from the CCF `HKParmInt` constituent for that HK parameter if an entry exists at all. Otherwise the value simply supplements the other validity ranges

assembled so far. The parameter names must be given in their long (see Sect. 3.4) form. If supplemental parameter names are to be given for which no long names are defined in the CCF constituent *HKParmInt* the corresponding short names can be used instead.

overrideparametervalues	no	string list	""	;-separated list of validity range expressions

This is the list of validity range expressions corresponding to the parameter names in `overrideparameters`. List items must be separated with the character ';' and each item must be a valid boolean selection expression as detailed in the documentation of package selectlib (see documentation of selectlib). The expression may contain the character '@' which will get replaced with the name of the corresponding HK parameter.

withgtiset	no	boolean	true	true|false

Boolean switch determining whether or not a GTI set should be generated. If set to `false`, for each exposure and applicable periodic HK file **hkgtigen** will print on `stdout` the command line with which **tabgtigen** will be invoked.

gtiset	no	string	gti.ds	name of non-existing data set

The name of the GTI data set to create.

timecolumn	no	string	TIME	name of existing column in HK file

Duplicates the parameter with the same name of task tabgtigen (see documentation of tabgtigen)

prefraction	no	real	0.5	0 <= `prefraction` <= 1

Duplicates the parameter with the same name of task tabgtigen (see documentation of tabgtigen)

postfraction	no	real	0.5	0 <= `postfraction` <= 1

Duplicates the parameter with the same name of task tabgtigen (see documentation of tabgtigen)

diagnosticmode	no	boolean	false	true|false

If set to `true` the task's diagnostic mode (Sect. 3.5) is entered.

diagfile	no	string	STDOUT	

The name of the ASCII diagnostic file to be generated in diagnostic mode (Sect. 3.5). The special value `STDOUT` is recognized as the standard output channel.

plotgtis	no	boolean	false	true|false

If set to `true` **dsplot** is invoked at the end to visualize the individual GTIs together with the total ones.

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.

SizeMismatch *(error)*
The number of HK parameter given in `overrideparameters` (Sect. 4) and the number of expressions in `overrideparametervalues` (Sect. 4) do not match.

NoSuchHKFile *(warning)*
The current ODF does not contain a HK files of a particular type.
corrective action: ignore

UnusedValidityRanges *(warning)*
An HK parameter name has been specified (either on the command line or in an *HKParmInt* entry) for which there is no corresponding column in any periodic HK file.
corrective action: ignore

GtiDataSetEmpty *(warning)*
The generated GTI data set is empty. Using it on any event list will result in no events getting selected.
corrective action: ignore

In addition all error and warning messages of the task **tabgtigen** and packages **cal**, **dal**, **oal** can occur.

6 Input Files

1. ODF periodic HK files for instrument specified via `instrument` parameter

7 Output Files

1. a GTI file (applicable to the entire observation period for the instrument specified via `instrument` parameter

8 Algorithm

```
subroutine hkgtigen
    + retrieve the list of validity ranges from CAL for instrument <inst>
    + merge/modify list according command line parameters => <valrangelist>
    + foreach periodic HK file <hk> in ODF for instrument <inst>
        + forach exposure <exp> of instrument <inst>
            + get list <l> of columns in <hk>
            + construct boolean expression <e> from <l> and <valrangelist>
            + invoke tabgtigen with <e>, on <file> => <gti>
```

```
      + merge all GTI sets <gti> constructed in inner loops
      + write final GTI set
end subroutine hkgtigen
```

9 Future developments

Possible future extensions are:

- Add a task parameter to control the time period to which the GTI data set is applicable, e.g. a single exposure instead of the entire observation. This will speed up the task's execution time but has no other substantial benefits.
- Support a more powerful syntax for the expressions in the `overrideparametervalues` (Sect. 4) parameter. E.g. `@1`, `@2`, etc. refer to the names of the first, second, ..., HK parameter given in `overrideparameters` (Sect. 4).

References

[1] ESA. XMM Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM). Technical Report XMM-SOC-ICD-0004-SSD Issue 2.5, ESA/SSD, June 2000. Found at the URL: `ftp://astro.estec.esa.nl/pub/XMM/documents/odf_icd.ps.gz`.

[2] ESA. Interface control document for the XMM current calibration file. Technical Report XMM-GEN-ICD-0005, ESA/SSD, Dec 2001. Issue 4.0.

[3] K. Galloway. XMM technical note: Periodic housekeeping telemetry definition. Technical Report XMM-SOC-TN-0040-SSD 0.2, ESA/SSD, August 5 1999.