rgsproc

June 2, 2019

Abstract

An interactive version of the RGS pipeline.

1 Instruments/Modes

Instrument	Mode
RGS	Spectroscopy
RGS	High Time Resolution
RGS	smallWindow

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

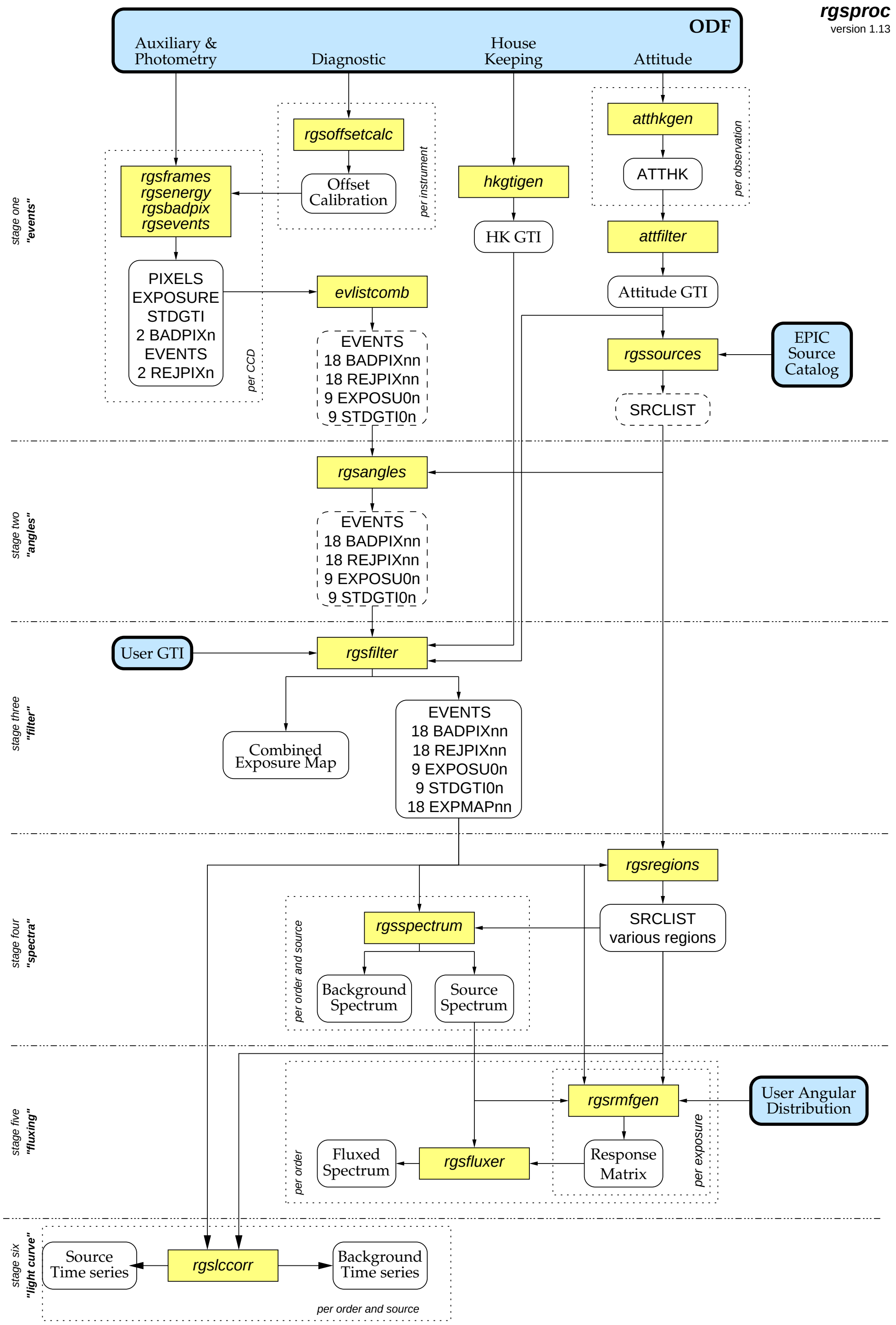

Figure 1: Diagram of **rgsproc** sub-tasks and data-flow. Each exposure is processed separately until the fluxing stage. Sub-tasks (yellow) appear within solid rectangles and process loops appear within dotted rectangles. Optional data files appear within rounded boxes, data files produced and used within the process are in green, with thick blue borders for permanent output files and dashed borders for incomplete files.

This task provides interactive control over the pipeline-style processing of RGS Spectroscopy and High Time Resolution (HTR) data, exposing the important options of the underlying tasks, while sparing the user the tedium and hazards of running each by hand. The user specifies the ODF directory as an environment variable or on the command line (see the OAL documentation). The resulting pipeline products and intermediate data files are created in the current working directory. By default all output files are named according to the convention of the Pipeline Processing Subsystem (PPS). Intermediate files that are discarded by the PPS—and so do not appear in the CD-ROM distribution—are distinguished by the use of lower case characters in the file name. **rgsproc** can redo later stages of processing without starting from scratch (parameters `entrystage` and `finalstage`) and without accessing the original ODF. The five processing stages, or entry points, are the organizing principle for the rest of this document:

1. **1:events**: preliminary tasks, source-independent calibrations
2. **2:angles**: aspect-drift corrections
3. **3:filter**: filter events and exposure
4. **4:spectra**: generate spectra
5. **5:fluxing**: generate response matrices and flux the spectra
6. **6:lightcurve**: generate light curves

rgsproc also allows the user to restrict the scope of processing to an enumerated subset of exposures within the observation (parameter `withinstexpids`), and an enumerated set of reflection orders (parameter `orders`). See Figure 1 for a block diagram of the data flow. The first stage of processing performs basic calibrations on the events in separate, CCD-specific *intermediate event lists* and then gathers them together in the *combined event list*. The second stage performs the source-specific aspect-drift correction and defines the channel grid for the events and exposure. The third stage performs the most basic filtering of the data—just to remove what is unusable—and generates exposure maps consistent with the filtering. The fourth stage produces spectra. The fifth stage generates response matrices and with them produces a combined, fluxed spectrum for the designated primary source and sixth stage generates source and background time series.

Note that fluxed spectra are inherently just a qualitative summary of the data, not to be used for quantitative analysis. Nevertheless the response matrices constructed in the fifth stage are large enough, by default, for meaningful use with spectral fitting packages such as XSPEC. At default size they take a very long time to compute, and are much larger than is necessary for the purpose of fluxing. Parameter `rmfbins`=250 produces very small matrices that are still quite adequate for fluxing, and much faster to compute, but such matrices should not be used for other purposes.

The combined event list takes from the intermediate event lists all of the events (combined into a single table), the collections of CCD-specific exposure tables and Good Time Interval (GTI) tables, and the collections of node-specific bad pixel tables and rejectable pixel tables. Name collisions among tables copied in from the intermediate event lists are avoided by placing the corresponding CCD number at the eighth character of the extension name. The following tables document the finished state of the combined event list. Next to the name of each table is the name of the task that created it, and for columns added later by a subsequent task the name of the responsible task is included among the comments. In addition to these and the nine `STDGTIOn` tables are eighteen node-specific exposure map image blocks, `EXPMAPnn`, created by **rgsfilter**.

In HTR mode the entire cross-dispersion dimension is collapsed into one row, inextricably mixing source and background events together. This mode is treated as a degenerate case of Spectroscopy in which the event coordinates are assigned randomly over the whole vertical extent of the chips. Likewise the exposure maps also cover the full cross-dispersion extent of the chips, but with variation only in the dispersion dimension. To annotate the minor structural differences between Spectroscopy and HTR processing, throughout this document a † marks items that do not apply to HTR mode data and a ‡ marks items that apply only to HTR mode data.

table: `EVENTS` — **evlistcomb**

BETA_CORR, XDSP_CORR	float32	aspect-drift corrected RGS angles — **rgsangles**
TIME	float64	time stamp of associated frame
FLAG	int32	status bits (boolean attributes)
BETA, XDSP	float32	uncorrected RGS angles
CHIPX, CHIPY	int16	chip-oriented pixel coordinates
PHA	int16	total uncalibrated pulse-height
SHAPE†	int8	DPP shape code (may be null)
GRADE†	int8	number of pixels combined
BETA_CHANNEL, XDSP_CHANNEL	int16	binned aspect-drift corrected RGS angles — **rgsangles**
PI	int16	calibrated pulse-height
CCDNR	int8	CCD number
SC_POINTING_RA, SC_POINTING_DEC, SC_POINTING_POS	real32	(optional) spacecraft pointing — **rgsangles**
M_LAMBDA	real32	(optional) nominal wavelength — **rgsangles**

table: `EXPOSU0n` — **rgsframes**

FRAME	int32	frame number
FLAG	int32	status bits (boolean attributes)
TIMEDEL	real32	integration time
TIME	real64	timestamp
FRACEXP0†, FRACEXP1†	real32	node-specific exposure corrections for pixels rejected on-board
ASPCDSP, ASPCXDSP	real32	aspect-drift correction parameters — **rgsangles**

table: `BADPIXnn` — **rgsbadpix**

CHIPX, CHIPY†	int16	chip-oriented pixel coordinates of column bottom
YEXTENT†	int16	upward length of bad column
TYPE	int16	bad pixel type code
BADFLAG	int16	bad pixel source identifier

table: `REJPIXnn` — **rgsevents**

FRAME	int32	frame number
FLAG	int32	status bits (boolean attributes)
CHIPX, CHIPY†	int16	chip-oriented pixel coordinates

The intermediate event lists contain some data of diagnostic value that is not carried into the combined event list. Except for this, there is no reason not to delete them, and accordingly they are deleted by default (parameter `expunge`). The diagnostic value of the intermediate files lies mostly in the `PIXELS` table, from which the events are reconstructed. In addition to that there are optional detector coordinate columns in the intermediate `EVENTS` table, which locate the event centroids to greater precision than is preserved in the combined event list. The finished state of these two tables is as follows.

table: `PIXELS` — **rgsframes**

`FRAME`	int32	frame number
`TELX`, `TELY`†	int16	telemetered pixel coordinates
`ENERGY`	int16	telemetered pulse-height
`CCDNODE`	int8	CCD node number
`SHAPE`†	int8	shape code (from telemetered SER)
`GRADE`†	int8	number of pixels (from telemetered SER)
`RAWX`, `RAWY`†	int16	node-oriented pixel coordinates
`PI`	real32	calibrated pulse-height — **rgsenergy**
`EVENT`†	int32	index to associated event — **rgsevents**
`RAWY`‡	attrib	nominal y-coordinate for all events

table: `EVENTS` (intermediate) — **rgsevents**

`FRAME`	int32	frame number
`CHIPX`, `CHIPY`†	int16	chip-oriented pixel coordinates
`BETA`, `XDSP`	real32	uncorrected RGS angles
`PI`	int32	calibrated pulse-height
`FLAG`	int32	status bits (boolean attributes)
`SHAPE`†	int8	DPP shape code (may be null)
`GRADE`†	int8	number of pixels combined
`PHA`	int32	total uncalibrated pulse-height
`TIME`	real64	timestamp of associated frame
`DETX`, `DETY`, `DETZ`	real32	(optional) detector coordinates

The remainder of this section summarizes the contribution from each sub-task. The reader should consult the referenced task descriptions for full details.

3.1 Stage One: "events"

The first processing stage creates the attitude-drift GTI, the housekeeping GTI, the source list, the intermediate event lists, and the unfinished combined event list. Nothing in this stage depends upon the choice of primary source, and there will rarely be any reason for the user to restart **rgsproc** here. In preliminary steps two exposure-independent data files, the attitude time-series and (optionally) the pulse-height offsets, are created; these are used only within this stage.

3.1.1 atthkgen

atthkgen creates the attitude time-series file. The same instance of this file is used for all exposures within the observation, and so this step is skipped if the file already exists in the current working directory. *To force construction of a fresh attitude time-series the user must manually delete this file before running **rgsproc**.*

3.1.2 attfilter

attfilter creates the attitude-drift GTI as a filtering of the attitude time-series over the duration of the exposure being processed. The median pointing within that window is first computed and then all excursions in pointing beyond the specified limit (parameter `driftlimit`) are rejected.

3.2 hkgtigen

hkgtigen analyzes the ODF housekeeping data to determine the basic set of housekeeping GTI for each instrument. Although not exposure-specific in nature, this step is nevertheless repeated for each exposure in order to avoid inconsistency if the user alters the calibration data before restarting **rgsproc** at this stage.

3.2.1 rgsoffsetcalc

It is expected that the offset for each CCD's pulse-height calibration will depend on the actual light background, which is a function of the aspect of the telescope. **rgsoffsetcalc** utilizes the simultaneously transmitted diagnostic mode data (the *Q-dumps*) and determines the offset per diagnostic dataset. These are written together with time stamps into a calibration data file, which must be explicitly presented to the CAL by each task that would have them override the static calibration data. The time tags allow for potential interpolations of drifts of the offsets during an exposure. These calibrations are computed separately for each instrument, but not for each exposure, and so this step is skipped if the file already exists in the current working directory. *To force construction of fresh offsets the user must manually delete this file before running **rgsproc**.* Providing these calibrations to **rgsenergy** is not recommended currently, and the default is to skip this step altogether (parameter `calcoffsets`=no).

3.2.2 rgssources

rgssources creates the source list and adds to it the `SRCLIST` table. The extraction region extensions are added in the fourth stage ("spectra") by **rgsregions**. The `SRCLIST` table is initialized with a variety of known or supposed source positions for which off-axis angles are calculated with respect to a particular reference pointing. One of these is selected to be the primary source (`PRIMESRC`), which is the nominal target source used later for the aspect-drift corrections and for fluxing in the final stage. If a user-defined source position is provided (parameter `withsrc`) it automatically becomes the primary source; otherwise **rgssources** selects the primary source on its own (see `primestyle`=auto in the task description). Selected sources are also imported from the EPIC source catalog, if provided (parameter `withepicset`). The default reference pointing is the one computed earlier by **attfilter** (parameter `attstyle`=expmedian), and it is not recommended to override this setting. Currently the aspect-drift corrections are computed using an approximate formula that assumes only small excursions from the reference pointing.

Do not restart **rgsproc** at this stage ("events") for the purpose of adding more user-defined sources to the `SRCLIST` table. Running the pipeline over from scratch is not necessary and besides the new source will be added to a fresh source list rather than the existing one. Instead, invoke **rgssources** directly to update the existing source list and then restart **rgsproc** at the second stage ("angles") after a change of primary source or at the fourth stage ("spectra") after adding new spectrum sources or changing the background region.

3.2.3 rgsframes

rgsframes creates the intermediate event lists and adds to them the `PIXELS` and `EXPOSURE` tables, and an internally generated GTI table called `STDGTI`. The ODF data are reformatted to some extent, and tested for faults and inconsistencies. The GTI table created here sets the most relaxed standard for what time intervals may be included in the data products, and subsequent stages can only act to further restrict these intervals.

3.2.4 rgsenergy

rgsenergy adds the PI column to the PIXELS table of the intermediate event lists. This is the calibrated energy value for each telemetered pixel.

3.2.5 rgsbadpix

rgsbadpix adds the two node-specific BADPIXn tables to the intermediate event lists. These tables are a combination of known bad pixels from the static calibration data and bad pixels found by analyzing the PIXELS table. The task parameters allow either contribution to be suppressed.

3.2.6 rgsevents

rgsevents adds the EVENTS and REJPIXn tables to the intermediate event lists along with an index of event associations in the PIXELS table. Multiple pixel events are recombined and the event centroid positions are computed. These positions are given in so-called "hardware" coordinates, which must be corrected for aspect drift before they can be used for spectral analysis. Various other event properties, such as contamination from bad pixels, are noted in the FLAG column. The REJPIXn tables contain the lists of pixels not already indicated in the BADPIXn tables that might later need to be excluded from the exposure maps.

3.2.7 evlistcomb

evlistcomb creates the combined event list from the set of intermediate event lists for one exposure. The EVENTS tables are merged and other tables, except for the PIXELS table, are copied whole, with the number of the originating CCD appended to the extension name to avoid name conflicts. The intermediate event lists are not required beyond this point.

3.3 Stage Two: "angles"

The second processing stage corrects the event coordinates for aspect drift and establishes the dispersion and cross-dispersion channel definitions, according to which the events and exposure are binned. This stage adds new attributes and columns to the EXPOSU0n and EVENTS tables of the combined event list. The aspect-drift corrections performed in this stage are specific to the chosen primary source. When spectra for other sources are constructed in the final stage, the inapplicability of these corrections introduces a systematic error, which is expected to be negligible in most cases. Nevertheless the user may wish to change the primary source (see task **rgssources**) and restart **rgsproc** here to obtain the most rigorously correct spectrum for a source other than the one originally designated as primary.

3.3.1 rgsangles

rgsangles adds the dispersion and cross-dispersion channel numbers for each event to the EVENTS table. This involves correcting the event coordinates for aspect drift, which is done with respect to the primary source position specified in the SRCLIST table. The corrected floating-point coordinates are also added to each event, and the computed constants of the correction formula are added to each frame for the later construction of the exposure maps. The "binning" of the floating-point coordinates into discrete channel numbers is controlled by the task parameters, and these channel definitions are documented as attributes of the `BETA_CHANNEL` and `XDSP_CHANNEL` columns of the `EVENTS` table.

3.4 Stage Three: "filter"

The third processing stage filters the combined event list to produce the filtered event list, which contains the collection of node-specific exposure maps. It also creates the separate combined exposure map. The filtering performed here is not general; it is based only on event flags and frame timestamps, and the purpose is simply to remove data that are not usable for any purpose of analysis. Combining the event filtering with exposure map construction in one step assures mutual consistency. The combined event list is not used beyond this point, and can be deleted (parameter `expunge`) with little risk that anything of value will be lost.

Typically the interactive user will wish to adjust the time-filtering of the data, which is done by manually constructing a supplemental GTI table (see task **gtibuild**) and restarting **rgsproc** here. If the original combined event list is missing (as, for example, in the PPS) the original filtered event list is acceptable as a substitute if no changes have been made to the original attitude and housekeeping GTI. In fact, **rgsproc** will automatically replace a missing combined event list by copying the filtered event list when it is started at this or the previous stage. *However, this behavior can lead to unexpected results.* **rgsproc** cannot determine whether the filtering already applied in the existing filtered event list is compatible with the filtering about to be applied. The result will always be a fully self-consistent and valid filtered event list, but it may have more events and exposure removed from it than the user is expecting. The **ExtraFiltering** warning message is issued whenever an automatic copy is performed, to remind the user that the result may not be the same as would be obtained using the original combined event list.

Note that the time filtering automatically includes the attitude-drift and housekeeping GTI computed in the first stage, *if these files are present in the current working directory.* Deleting these files is therefore one way to disable them when it appears that something may be wrong with the housekeeping or attitude data. Although these files do not exist in a PPS distribution, this is not really a problem because the combined event list does not exist either and applying them to the filtered event list would be redundant.

3.4.1 rgsfilter

rgsfilter creates the filtered event list as an exact replica of the combined event list, then removes unwanted frames and events, and adds the node-specific exposure maps. As a default option, it also creates the combined exposure map as a separate file. The common practice of including the effective area in the exposure map is rejected because the effective area is handled more correctly by including it in the response matrix. The exposure maps are constructed on the channel space established previously by **rgsangles**, though each node-specific map is cropped to the exposed area of its particular node. The combined map covers the full channel space, so that channel numbers assigned to the events are coordinates in this map. The combined map is purely an end-product—it is not read by **rgsproc** or any of its sub-tasks. The filtering options provided by **rgsfilter** are very limited: events can be filtered on an enumerated subset of the `FLAG` bits (parameter `rejflags`), and frames can be filtered by their timestamp (parameter `auxgtitables`). The filtered event list holds a merged record of all flags and time intervals that have been removed from it.

3.5 Stage Four: "spectra"

The fourth processing stage creates spectra for various sources at each specified reflection order. The source list is completed by adding all the required extraction regions.

3.5.1 rgsregions

rgsregions adds the source and background extraction region extensions to the source list. For each relevant source position there is one region in the spatial plane and one region per (specified) reflection order in the energy plane. The relevant source positions refer to the sources marked for processing. The spatial background region covers the entire exposed surface of the focal plane, except for a buffer zone around the position of each source marked for exclusion. Note that the source selections (for processing and exclusion) are determined by the `procsrcsexpr` and `exclsrcsexpr`[†] parameters and recorded in the `SRCLIST` table as selection flags. To sanction the existing settings of these flags supply empty strings for these parameters. A spatial background region cannot be defined for HTR mode data; there is instead a set of order-specific background regions in the energy plane for each selected source.

3.5.2 rgsspectrum

rgsspectrum creates a source and background spectrum for each selected source in each of the specified reflection orders. And also, you can create a background model spectrum, using the parameter `withbackgroundmodel` (see **rgsbkgmodel** documentation). The source spectrum is not background-corrected by default (parameter `bkgcorrect`). The background spectra is not used by this or the following stage, but it can be used for a later analysis, and therefore is produced by default (parameter `withbkgset`). Note that the format of the spectra conforms to the new OGIP standard, which is supported as of LHEASOFT Version 5.1 (XSPEC Version 11.1). This is a spectrum of counts with the exposure time correction and background scaling factor recorded separately for each channel.

if `withheliocorr` is enabled, the task **heliocentriccorr** is executed for all spectra, writing the heliocentric velocity keyword (VHELICOR) in the spectrum header.

3.6 Stage Five: "fluxing"

The fifth processing stage creates the fluxed spectrum of the primary source. A necessary by-product of this stage is a set of order-specific response matrices for each exposure processed. While a fluxed spectrum is never appropriate for quantitative analysis (it is just a qualitative summary of the data), the response matrices produced at the default settings are adequate for use with external fitting packages.

3.6.1 rgsrmfgen

rgsrmfgen generates the response matrices. These are required by **rgsfluxer** for fluxing spectra, but their true value is seen in such sophisticated analysis packages as XSPEC, where a resolution of 3000 to 5000 incident energy bins (parameter `rmfbins`) is recommended for good results. Unfortunately such large matrices take a lot of time to compute—massive overkill if used only for fluxing. The default resolution is set high to avoid misleading the unwary, but lowering it to `rmfbins`=250 is highly recommended when fluxing is the only use intended. Similarly, the default for parameter `fftdim` is adequate for any purposes, but `fftdim`=1 is faster and not too crude for fluxing (this parameter controls losses due to truncation at the wings of various distributions used in computing the response). The response matrices produced in this stage apply only to the primary source. To make the response matrices appropriate for analyzing an extended source, a customized LSF is required (parameters `withangdist` and `withmirrorpsf`).

3.6.2 rgsfluxer

rgsfluxer combines the primary source spectrum for each exposure and specified instrument with the corresponding response matrices to produce a single fluxed spectrum for the whole observation. By default the various reflection orders are not fluxed together (parameter `mergeorders`), yielding one fluxed spectrum per specified reflection order. No popular standardized format exists for fluxed spectra, and the task parameters provide a variety of options. Corrections are made for the background when necessary.

3.7 Stage Six: “lightcurve”

The sixth processing stage creates source and background time series for each exposure and order using **rgslccorr**.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

<table><tr><td><b>entrystage (global)</b></td><td>no</td><td>choice</td><td>1:events</td><td>1:events | 2:angles | 3:filter | 4:spectra | 5:fluxing | 6:lightcurve</td></tr></table>

The stage at which to begin processing.

<table><tr><td><b>finalstage (global)</b></td><td>no</td><td>choice</td><td>5:fluxing</td><td>1:events | 2:angles | 3:filter | 4:spectra | 5:fluxing | 6:lightcurve</td></tr></table>

The last stage to process.

withinstexpids (global)	no	boolean	false	

On true: only the specified list of exposures (parameter `instexpids`) are processed. On false: all Spectroscopy exposures are processed.

instexpids (global)	no	list of strings		

List of instrument-exposure identifiers. These take two possible forms: a six-character string such as “`R1S001`” or the full name of any ODF file related to the desired exposure. *Requires* `withinstexpids=true`.

orders (global)	no	list of integers	1 2	$1 \leq integer \leq 6$

Which reflection orders to process in the final two stages (“spectra” and “fluxing”).

withprefix (global)	no	boolean	false	

A prefix string begins the name of every file written to or read from the current working directory. On true: the specified prefix string (parameter `prefix`) is enabled. On false: the PPS prefix string is inferred from the ODF.

prefix (global)	no	string		

Filename prefix string used in the current working directory. *Requires* `withprefix=true`.

expunge (global)	no	list of choices	events	events | merged

Which intermediate datasets may be deleted automatically.

spectrumbinning	no	choice	lambda	lambda beta

Spectrum binning type. Used to accumulate exposure maps and spectrum. Also used to calculate regions

timestep (Stage1/atthkgen)	no	real	1	$real \geq 0$

Time resolution in seconds on which to construct the spacecraft attitude time-series file.

driftlimit (Stage1/attfilter)	no	real	5	$real \geq 0$

Threshold in arc minutes for filtering out time intervals in which the spacecraft attitude drifted too far from the median.

onlylastatt (Stage1/attfilter)	no	boolean	false	

If true only the last attitude points (100) are taken for calculating (median) the reference attitude for filtering.

calcoffsets (Stage1/rgsoffsetcalc)	no	boolean	false	

The energy calibrations require pulse-height offsets from the CCF. Override values for these can be computed from the diagnostic data. On true: offset overrides are computed and passed to **rgsenergy**. On false: offset overrides are not computed.

withoffsethistogram (Stage1/rgsoffsetcalc)	no	boolean	false	

On true: the pulse-height distributions from the offset computation are written to an output dataset. On false: the distributions are discarded. *Requires* `calcoffsets=true`.

withepicset (Stage1/rgssources)	no	boolean	false	

On true: suitable entries from the specified EPIC catalog of detected sources (parameter `epicset`) are added to the output `SRCLIST` table. On false: no EPIC sources are added.

epicset (Stage1/rgssources)	no	dataset		

An EPIC source list, such as produced by tasks **emldetect** and **eboxdetect**. *Requires* `withepicset=true`.

withsrc **(Stage1/rgssources)**	no	boolean	false	

On true: the specified source (parameters `srclabel`, *etc.*) is added to the output `SRCLIST` table and is established as the primary source for subsequent analyses. On false: there is no user-defined source and so **rgssources** selects a primary source on its own, giving preference to the Observation Proposal.

srclabel **(Stage1/rgssources)**	no	string	USER	

User-defined source: `LABEL` column value. This is the name of the source; ideally it should be meaningful and convenient. *Requires* `withsrc=true`.

srcrate **(Stage1/rgssources)**	no	real	0	$real \geq 0$

User-defined source: `RATE` column value. This is the RGS band brightness of the source in counts per second, and is used to compute relative confusion between the sources. *Requires* `withsrc=true`.

srcstyle **(Stage1/rgssources)**	no	choice	radec	radec | wrtatt

User-defined source: this parameter specifies how the source position is to be defined. radec: celestial coordinates (parameters `srcra` and `srcdec`). wrtatt: RGS angular coordinates (parameters `srcdisp` and `srcxdsp`). This choice is made persistent through the `FIXED_ON_SKY` column (set true for celestial coordinates) so that subsequent changes to the reference spacecraft attitude affect the derived coordinates but not the defining coordinates. *Requires* `withsrc=true`.

srcra **(Stage1/rgssources)**	no	real	-999	

User-defined source: `RA` column value. This is the right ascension in decimal degrees. *Mandatory when applicable. Requires* `srcstyle=radec`.

srcdec **(Stage1/rgssources)**	no	real	-999	

User-defined source: `DEC` column value. This is the declination in decimal degrees. *Mandatory when applicable. Requires* `srcstyle=radec`.

srcdisp **(Stage1/rgssources)**	no	real	0	

User-defined source: `DELTA_DISP` column value. This is the dispersion-direction offset from the pointing axis, in arc minutes. *Requires* `srcstyle=wrtatt`.

srcxdsp **(Stage1/rgssources)**	no	real	0	

User-defined source: `DELTA_XDSP` column value. This is the cross-dispersion-direction offset from the pointing axis, in arc minutes. *Requires* `srcstyle=wrtatt`.

attstyle **(Stage1/rgssources)**	no	choice	expmedian	expmedian | start | user

Specifies the method to use for defining the reference spacecraft attitude (pointing of the instrument). expmedian: calculated from the attitude time-series file. start: start-of-exposure attitude as given in the ODF. user: enables the specified attitude (parameters `attra`, `attdec` and `attapos`).

attra **(Stage1/rgssources)**	no	real	-999	

User-defined attitude: right ascension in decimal degrees. *Mandatory when applicable. Requires* `attstyle=user`.

attdec **(Stage1/rgssources)**	no	real	-999	

User-defined attitude: declination in decimal degrees. *Mandatory when applicable. Requires* `attstyle=user`.

attapos (Stage1/rgssources)	no	real	-999	

User-defined attitude: position angle in decimal degrees. *Mandatory when applicable. Requires* `attstyle=user`.

withdiagoffset (Stage1/rgsenergy)	no	boolean	false	

On true: files included in the ODF (from 2004 on) contain averages of diagnostic images over three consecutive orbits around the data. They are used in this case for an offset subtraction pixel by pixel.

withgain (Stage1/rgsenergy)	no	boolean	true	

On true: calibration of the telemetered energy values includes the gain correction. On false: gain correction is not performed.

withcti (Stage1/rgsenergy)	no	boolean	true	

On true: calibration of the telemetered energy values includes the CTI correction. On false: CTI correction is not performed.

badpixalgo (Stage1/rgsbadpix)	no	choice	badpixalgo	badpixalgo | embadpixalgo

User-defined bad pixels and column algorithm: this parameter specifies the algorithm to be used to determine bad pixels and columns.

embadpixalgo	no	boolean	no	

Boolean parameter to enable the call the embadpixfind task to find bad pixels and columns.

keepcool	no	boolean	yes	

If set to "no", columns showing often larger CTI will be flagged (taken also from the CCF).

withadvisory (Stage1/rgsbadpix)	no	boolean	true	

The `BADPIXnn` tables are a compilation of all known bad pixel locations. The choice of how to use this information to filter the data is not made until the third stage of processing ("filter"). On true: advisory (non-uplinked) bad pixels from the CCF are included. On false: no advisory bad pixels.

withfoundhot (Stage1/rgsbadpix)	no	boolean	true	

On true: the telemetry is analyzed for pixels that appear to be hot (frequently yielding spurious detections) according to criteria set by the following parameters. Any such pixel locations are then included in the `BADPIXnn` tables. On false: hot pixel finding is disabled. *Not applicable to HTR mode data.*

pixnoiselimit (Stage1/rgsbadpix)	no	integer	0	$0 \leq integer \leq 4095$

Hot pixel finding: lowest uncalibrated single-pixel energy to be considered. *Requires* `withfoundhot=true`.

pixsharpness (Stage1/rgsbadpix)	no	real	5	$real \geq 0$

Hot pixel finding: sharpness criterion (see **rgsbadpix** documentation). *Requires* `withfoundhot=true`.

colnoiselimit (Stage1/rgsbadpix)	no	integer	250	$0 \leq integer \leq 4095$

Hot column finding: lowest uncalibrated single-pixel energy to be considered. *Requires* `withfoundhot=true`.

colsharpness (Stage1/rgsbadpix)	no	real	8	$real \geq 0$

Hot column finding: sharpness criterion (see **rgsbadpix** documentation). *Requires* `withfoundhot=true`.

reconstruct (Stage1/rgsevents)	no	boolean	true	

On true: spatially adjacent events in the telemetered data are combined into a single “reconstructed” event. On false: no event reconstruction. *Not applicable to HTR mode data.*

withdetcoord (Stage1/rgsevents)	no	boolean	false	

On true: in addition to chip coordinates and angular coordinates, detector coordinates for each event are added to the `EVENTS` table. Parameter `detcoord` selects among the available coordinate systems. On false: detector coordinates are omitted.

detcoord (Stage1/rgsevents)	no	choice	cam	chip | cam | sac

Specifies a standard XMM coordinate system for the output detector coordinates. chip: CHIPCOORD with sub-pixel accuracy. cam: CAMCOORD1. sac: SACCOORD. *Requires* `withdetcoord=true`.

betabinning (Stage2/rgsangles)	no	choice	binSize	Range | binSize

Specifies how the dispersion channels are to be defined. Range: interval to be subdivided (parameters `betamin` and `betamax`) and number of channels. binSize: midpoint and width of the first channel (parameters `betabinref` and `betabinwidth`) and number of channels.

betamin (Stage2/rgsangles)	no	real	3×10^{-2}	

Lower boundary of the first dispersion channel, in radians. *Requires* `betabinning=Range`.

betamax (Stage2/rgsangles)	no	real	8×10^{-2}	

Upper boundary of the last dispersion channel, in radians. *Requires* `betabinning=Range`.

betabinref (Stage2/rgsangles)	no	real	3.578524×10^{-2}	

Midpoint of the first dispersion channel, in radians. *Requires* `betabinning=binSize`.

betabinwidth (Stage2/rgsangles)	no	real	1.208×10^{-5}	*real* ≥ 0

Dispersion channel width in radians. *Requires* `betabinning=binSize`.

nbetabins (Stage2/rgsangles)	no	integer	3400	*integer* ≥ 1

Number of dispersion channels.

lambdabinref	no	r	4.005	

midpoint of bin for which LAMBDACHA is one [Angstrom]

lambdabinwidth	no	r	0.010	

width of lambda bin [Angstroms]

nlambdabins	no	i	3600	> 0

number of lambda bins.

xdispbinning (Stage2/rgsangles)	no	choice	binSize	Range | binSize

Specifies how the cross-dispersion channels are to be defined. Range: interval to be subdivided (parameters `xdispmin` and `xdispmax`) and number of channels. binSize: midpoint and width of the first channel (parameters `xdispbinref` and `xdispbinwidth`) and number of channels.

xdispmin (Stage2/rgsangles)	no	real	-1×10^{-3}	

Lower boundary of the first cross-dispersion channel, in radians. *Requires* `xdispbinning=Range`.

xdispmax (Stage2/rgsangles)	no	real	1×10^{-3}	

Upper boundary of the last cross-dispersion channel, in radians. *Requires* `xdispbinning=Range`.

xdispbinref (Stage2/rgsangles)	no	real	-9.126×10^{-4}	

Midpoint of the first cross-dispersion channel, in radians. *Requires* `xdispbinning=binSize`.

xdispbinwidth (Stage2/rgsangles)	no	real	1.08×10^{-5}	*real* ≥ 0

Cross-Dispersion channel width in radians. *Requires* `xdispbinning=binSize`.

nxdispbins (Stage2/rgsangles)	no	integer	170	*integer* ≥ 1

Number of cross-dispersion channels.

withpointingcolumn (Stage2/rgsangles)	no	boolean	false	

On true: the spacecraft attitude (pointing of the instrument) for each event is added to the `EVENTS` table. On false: the attitude is omitted.

withmlambdacolumn (Stage2/rgsangles)	no	boolean	false	

On true: the primary source $m\lambda$ for each event is added to the `EVENTS` table. On false: the $m\lambda$ is omitted.

withheliocentriccorr	no	b	Yes	

Corrects $m * \lambda$ column from heliocentric velocity

withsunanglecorr	no	b	Yes	

Corrects $m * \lambda$ column from Sun angle correction

auxgtitables (Stage3/rgsfilter)	no	list of tables		

List of OGIP compliant auxiliary GTI tables for filtering the event list in conjunction with its internal `STDGTIOn` tables.

withcombmap (Stage3/rgsfilter)	no	boolean	true	

rgsfilter produces a filtered exposure map separately for each node, as required by various tasks in the following stages, but a combined map is more convenient to look at and has various uses. On true: the combined exposure map is written to an output dataset. On false: no combined exposure map.

driftbinsize (Stage3/rgsfilter)	no	real	1	

Size in arcseconds of the square bins used to group frames according to their aspect drift in dispersion and cross-dispersion. Grouping improves the speed of constructing the exposure maps, at a slight cost in accuracy.

rejflags (Stage3/rgsfilter)	no	list of choices	BAD_SHAPE ON_NODE_INTERFACE ON_BADPIX NEXT_TO_BADPIX ON_WINDOW_BORDER BELOW_ACCEPTANCE	BAD_SHAPE | ON_NODE_INTERFACE | ON_BADPIX | NEXT_TO_BADPIX | ON_WINDOW_BORDER | BELOW_ACCEPTANCE

List of rejection flags for filtering the event list.

spatialres (Stage4/rgsregions)	no	integer	5	*integer* ≥ 2

Dispersion-axis spacing of vertices for image regions and the background region, specified as the number of vertices across a node. *Not applicable to HTR mode data.*

orderres (Stage4/rgsregions)	no	integer	3	*integer* ≥ 2

Dispersion-axis spacing of vertices for order regions, specified as the number of vertices across a node.

xpsfincl (Stage4/rgsregions)	no	real	90	$0 \leq real \leq 100$

Size of image regions, specified by the percentage of the cross-dispersion PSF covered between each pair of vertices with the same dispersion coordinate. *Not applicable to HTR mode data.*

xpsfexcl (Stage4/rgsregions)	no	real	95	$0 \leq real \leq 100$

Size of the exclusion regions in the background region, specified by the percentage of the cross-dispersion PSF covered between each pair of vertices with the same dispersion coordinate and associated source. *Not applicable to HTR mode data.*

pdistincl (Stage4/rgsregions)	no	real	90	$0 \leq real \leq 100$

Size of order regions, specified by the percentage of the pulse-height distribution covered between each pair of vertices with the same dispersion coordinate.

procsrcsexpr **(Stage4/rgsregions)**	no	string	INDEX==#PRIMESRC	

Selection expression indicating which sources to process for image and order regions. The default specifies the primary source only. When restarting **rgsproc** on its own output, leave this parameter blank to accept the selections already in place.

exclsrcsexpr **(Stage4/rgsregions)**	no	string	INDEX==#PRIMESRC	

Selection expression indicating which sources to exclude from the background region. The default specifies the primary source only. When restarting **rgsproc** on its own output, leave this parameter blank to accept the selections already in place. *Not applicable to HTR mode data.*

bkgcorrect **(Stage4/rgsspectrum)**	no	boolean	no	

On true: background-correction is applied to all source spectra. On false: no background correction.

withbkgset **(Stage4/rgsspectrum)**	no	boolean	true	

On true: a background spectrum is produced for each source spectrum. On false: no background spectra.

rebin **(Stage4/rgsspectrum)**	no	integer	1	*integer* ≥ 1

Channel rebinning factor. The events and exposure are rebinned going into the histogram, which gives a more stable result than if the full-resolution spectrum is rebinned later (a very minor point).

withfracexp **(Stage4/rgsspectrum)**	no	boolean	no	

Enables addition of the non-standard (but harmless) `FRAC_EXP` column to the output spectra. This is the fraction of full exposure in each channel.

withbackgroundmodel **(Stage4/rgsspectrum)**	no	boolean	no	

On true: a background template model spectrum is produced. See **rgsbkgmodel** for details. On false: no background template model is produced.

exposed **(Stage4/rgsspectrum)**	no	real	0.1	0-1

Channels with less than this fraction of full exposure are marked as bad in the output `QUALITY` column.

edgechannels **(Stage4/rgsspectrum)**	no	integer	2	*integer* ≥ 0

The number of channels at the edges of each chip to be marked as bad in the output `QUALITY` column. For reasons not well understood at this time, the two channels at either edge of every chip do not seem to be properly calibrated.

badquality **(Stage4/rgsspectrum)**	no	integer	1	0-5

The `QUALITY` column value used to indicate a bad channel. The default causes bad channels to be discarded automatically by XSPEC. Advanced users of XSPEC may prefer to change this for diagnostic purposes.

withheliocorr (Stage4/rgsspectrum)	no	boolean	no	

On true: heliocentriccorr task is executed, writing the VHELICOR keyword in the spectrum header.

witharffile (Stage5/rgsrmfgen)	no	boolean	no	

On true: rgsrmfgen will create two independent ARF and RMF files.

rmfbins (Stage5/rgsrmfgen)	no	integer	4000	$integer \geq 1$

Number of rows (incident-energy bins) in the response matrices: effectively the energy resolution. The columns match the channel space of the spectra. The default is fine for general purposes, but very slow to compute; 250 is adequate when only fluxing is intended.

fftdim (Stage5/rgsrmfgen)	no	integer	3	$1 \leq integer \leq 5$

The wings of the various distributions that are convolved together to form the narrow features of the line-spread function are truncated to limit the size of the convolution space. An increment of one in this parameter doubles the size of the convolution space; the larger the convolution space, the slower the computation and the less power is lost to the truncation. The default is fine for general purposes, but 1 is adequate when only fluxing is intended.

withmirrorpsf (Stage5/rgsrmfgen)	no	boolean	yes	

On true: the standard mirror PSF distribution from the CAL is included in the LSF. On false: the mirror PSF is not included. The option to disable the standard mirror PSF is provided primarily for use with extended sources and a custom angular distribution that implicitly contains the mirror PSF.

withdynefferea (Stage5/rgsrmfgen)	no	boolean	yes	

On true: execute rgsrmfgen using the dynamic effective area correction based on Chebyshev parameters.

withrectification	no	boolean	no	

Use empirical RGS effective area correction

witheffectiveareacorrection	no	boolean	no	

Use RGS effective area correction

withangdist (Stage5/rgsrmfgen)	no	boolean	no	

On true: the user-defined angular distribution is included in the LSF. On false: it is not included. This option is provided for the purposes of extended-source analysis, but the imaginative user may find other applications for it.

angdistset (Stage5/rgsrmfgen)	no	file	angdist.txt	

An ASCII formatted distribution in RGS dispersion offset angle, for use in generating a customized LSF, presumably for an extended source. See **rgsrmfgen** for details on the file format. *Requires* `withangdist=true`.

mergeorders (Stage5/rgsfluxer)	no	boolean	false	

On true: all the spectra (both instruments, all exposures and reflection orders) are combined together in a single fluxed spectrum. On false: a separate fluxed spectrum is produced for each reflection order.

flxformat (Stage5/rgsfluxer)	no	choice	dal	dal | ascii | qdp

Specifies the format of the output fluxed spectrum. dal: FITS-encoded SAS dataset. ascii: basic ASCII-encoded text file. qdp: text file formatted as input to qdp.

flxmode (Stage5/rgsfluxer)	no	choice	wavelength	wavelength | energy

Specifies the units for defining the output fluxing bins (parameters `flxmin` and `flxmax`). wavelength: units of Å. energy: units of keV.

flxmin (Stage5/rgsfluxer)	no	real	4	*real* ≥ 0

Lower edge of the first output bin in the units set by `flxmode`. *The default is not appropriate for* `flxmode=energy`.

flxmax (Stage5/rgsfluxer)	no	real	39	*real* ≥ 0

Upper edge of the last output bin in the units set by `flxmode`. *The default is not appropriate for* `flxmode=energy`.

flxbins (Stage5/rgsfluxer)	no	integer	3400	*integer* ≥ 1

Number of output bins.

flxquality (Stage5/rgsfluxer)	no	list of integers	0	$0 \leq$ *integer* ≤ 5

This parameter controls how the quality flags from the input spectrum files are propagated into the output fluxed spectrum. If the quality flag of an input spectral channel is not represented in this list, the associated count rate is simply discarded. Where multiple channels, either from the same or from different spectra, overlap the same output bin, the output quality is degraded to the worst input quality. This ranking is set by the order of the list itself: from highest quality to lowest. Thus the list "0 2 1" preserves all input channels and degrades the quality in the sequence, "good" to "dubious" to "bad". The default preserves only the "good" channels. When only one quality value is permitted, the quality column is omitted from the output file. Quality flags three, four and five are defined within XSPEC, but are not currently used by **rgsspectrum**.

withflxnan (Stage5/rgsfluxer)	no	boolean	false	

Undefined flux values naturally occur at spectral channels where there is no exposure, such as in the gaps between chips, and are represented by NaN (IEEE standard encoding). Sadly, many standard analysis packages do not react well to this. On true: all instances of NaN in the output are replaced with the specified finite alternative (parameter `flxnan`). On false: NaN is permitted in the output.

flxnan (Stage5/rgsfluxer)	no	real	0	

Value used instead of NaN (or "no", in the case of `flxformat=qdp`) to indicate an undefined value in the output fluxed spectrum. *Requires* `withflxnan=true`.

timebinsize	no	1000	real	> 0

Size of time bins.

withbkgsubtraction	no	bool	no	

Enable background subtraction

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.

InvalidStageRange *(error)*
The parameter `finalstage` must not be set to an earlier stage of processing than the parameter `entrystage`.

InternalError *(error)*
An internal inconsistency in the program logic has been encountered. In some cases this may indicate that the parameter configuration file has been corrupted.

SkippedExposure *(warning)*
There are no events to process.
corrective action: Further processing of the empty exposure is aborted. Processing resumes with the next exposure.

ExtraFiltering *(warning)*
The filtered event list has been used as a substitute for the combined event list required by tasks **rgsangles** and **rgsfilter**.
corrective action: The required combined event list file is created by duplicating the existing filtered event list file. Whatever filtering had previously been applied is thereby retained

in the new filtered event list. If that is not what the user desires the only alternative is to reconstruct the original combined event list by running **rgsproc** over from the first stage.

SkippedSpectrum *(warning)*
The primary source spectrum for a requested exposure cannot be found under the expected name.
corrective action: The missing spectrum is not included in the fluxed spectrum. In the case of a single exposure, no fluxed spectrum is produced.

SmallWindowMode(rgsspectrum) *(warning)*
Small window mode detected on a particular exposure. In this case the background generation and the background subtraction is skipped in rgsspectrum.
corrective action:

Small_Window_Mode(rgsfluxer) *(warning)*
Small window mode detected on a particular exposure. In this case, rgsfluxer and rgslccorr tasks do not include the background for those exposures.
corrective action:

LambdaCorrectionsDisabled *(warning)*
RGS Processing requested in Beta space. Lambda corrections (Heliocentric, Sun aspect angle will be disabled) will be automatically disabled.
corrective action:

Other errors and warnings may be produced by sub-tasks.

6 Input Files

- an ODF containing RGS data
- an optional OGIP compliant GTI file
- an optional EPIC source list as specified for **rgssources**
- an optional angular distribution

7 Output Files

- *<prefix>*`0BX000ATTTSR0000.FIT`

  Attitude time-series data. One file for the entire observation. See task **atthkgen**.

- *<prefix>*`R`*<rgs>*`X000offset0000.FIT`

  Calibration override file for the pulse-height offsets, produced if `calcoffsets` is true (non-default) and there is diagnostic mode data in the ODF. One file for each RGS instrument. See task **rgsoffsetcalc**.

- *<prefix>*`R`*<rgs>*`X000offhst0000.FIT`

  Histogram of the diagnostic mode data, produced if both `calcoffsets` and `withoffsethistogram` are true (non-default) and there is diagnostic mode data in the ODF. One file for each RGS instrument. See task **rgsoffsetcalc**.

- *<prefix>*R*<rgs><exposure>*`attgti0000.FIT`

  Attitude-drift GTI table generated from the attitude time-series. One file for each exposure. See task **attfilter**.

- *<prefix>*R*<rgs><exposure>*`hkgti_0000.FIT`

  Housekeeping GTI table. One file for each exposure. See task **hkgtigen**.

- *<prefix>*R*<rgs><exposure>*`events`*<ccd>*`000.FIT`

  Intermediate event list, created by **rgsframes** and last modified by **rgsevents**. One file for each CCD, for each exposure. Contains the following extensions: `PIXELS`, `EXPOSURE`, `STDGTI`, `BADPIXn`, `EVENTS`, `REJPIXn`.

- *<prefix>*R*<rgs><exposure>*`SRCLI_0000.FIT`

  List of target sources and associated extraction regions. Created by **rgssources** and last modified by **rgsregions**. One file for each exposure.

- *<prefix>*R*<rgs><exposure>*`merged0000.FIT`

  Combined event list (unfiltered), created by **evlistcomb** and last modified by **rgsangles**. Contains the following extensions: `EVENTS`, `BADPIXnn`, `REJPIXnn`, `EXPOSU0n`, `STDGTI0n`. One file for each exposure.

- *<prefix>*R*<rgs><exposure>*`EVENLI0000.FIT`

  Filtered event list, containing the following extensions: `EVENTS`, `BADPIXnn`, `REJPIXnn`, `EXPOSU0n`, `STDGTI0n`, `EXPMAPnn`. One file for each exposure. See task **rgsfilter**.

- *<prefix>*R*<rgs><exposure>*`EXPMAP0000.FIT`

  Combined exposure map, produced if `withcombmap` is true. One file for each exposure. See task **rgsfilter**.

- *<prefix>*R*<rgs><exposure>*`SRSPEC`*<order><source>*`.FIT`

  Source spectrum. One file for each specified order, for each selected source, for each exposure. See task **rgsspectrum**.

- *<prefix>*R*<rgs><exposure>*`BGSPEC`*<order><source>*`.FIT`

  Background spectrum. One file for each specified order, for each selected source, for each exposure. See task **rgsspectrum**.

- *<prefix>*R*<rgs><exposure>*`RSPMAT`*<order><source>*`.FIT`

  Response matrix. One file for each specified order, for each exposure. See task **rgsrmfgen**.

- *<prefix>*`OBX000fluxed`*<order>*`000.`*<format>*

  Fluxed spectrum, produced if `mergeorders` is false. One file for each specified order. See task **rgsfluxer**.

- *<prefix>*`OBX000fluxed0000.`*<format>*

  Fluxed spectrum, produced if `mergeorders` is true. One file for the entire observation. See task **rgsfluxer**.

8 Algorithm

```
========== stage one: "events" ==========

FOR EACH specified exposure

  IF (!exists(atttsr))   atthkgen: create(atttsr)

  IF (calcoffsets && !exists(offset))   rgsoffsetcalc: create(offset,[offhst])

  hkgtigen: create(hkgti_)
  attfilter: read(atttsr) create(attgti)
  rgssources: read(attgti) create(srcli_)

  FOR EACH ccd

    rgsframes: create(events)
    rgsenergy: read([offset]) modify(events)
    rgsbadpix: modify(events)
    rgsevents: modify(events)

  evlistcomb: read(EACH events) create(merged)

========== stage two: "angles" ==========

  if (!exists(merged) && exists(evenli))   clone: read(evenli) create(merged)

  rgsangles: read(srcli_) modify(merged)

========== stage three: "filter" ==========

  if (!exists(merged) && exists(evenli))   clone: read(evenli) create(merged)

  rgsfilter: read(merged,attgti,hkgti_,auxgtitables) create(evenli,[expmap])

========== stage four: "spectra" ==========

  rgsregions: read(evenli) modify(srcli_)

  FOR EACH selected source
    FOR EACH specified order

      rgsspectrum: read(evenli,srcli_) create(srspec,[bgspec])
      rgsbkgmodel: read(evenli) create([bkgmodelspec])

========== stage five: "fluxing" ==========

FOR EACH specified order
  FOR EACH specified exposure

    rgsrmfgen: read(evenli,srcli_,srspec,[angdist]) create(rspmat)

  IF (!mergeorders) rgsfluxer: read(EACH rspmat,EACH srspec) create(fluxed)

IF (mergeorders) rgsfluxer: read(EACH rspmat,EACH srspec) create(fluxed)
```

9 Comments

References