The MATHPHA User's Guide

Ian M George
Code 668,
NASA/GSFC,
Greenbelt, MD20771

Version: 1995 Aug 23
MATHPHA v4.0.0

SUMMARY

`MATHPHA` is a task within the HEASARC sub-package
of FTOOLS which allows users to perform mathematical operations on PHA
datasets. Here we describe in detail the required user inputs, available com-
mands & current limitations of the task. A number of examples are also
provided for illustration.

LOG OF SIGNIFICANT CHANGES

Release Date	Sections Changed	Brief Notes
1995 Mar 30		Original Version
1995 Apr 14	All	Update for `MATHPHAv3.5.0`
1995 Aug 23	3, 8	Update for `MATHPHAv4.0.0`

1 INTRODUCTION

`MATHPHA` takes a user-defined input expression representing mathematical operations to be performed on (one or more) PHA datasets. The input expression is thus allowed to contain PHA filenames (and optionally, an extension number), a number of mathematical operators, integers, reals, and pairs of curved brackets (”(” & ”)”). Assuming the expression is able to be parsed (and a number of other criteria are satisfied), the operations specified are performed on the PHA dataset, the errors propagated (see Section 3), and a new file containing the resultant PHA dataset is written.

The units in which the algebra is performed is controlled by a parameter (`units`; see Section 2), the default of which is COUNTS (as opposed to COUNTS PER SECOND). This is true **IR-RESPECTIVE** of whether the original PHA data themselves are stored as counts or counts per second. The exposure time written for the new PHA dataset is controlled by a user-defined parameter (`exposure`; see Section 4). Users also have the option of specifying the values written to the o/p file of several other mandatory keywords (see Section 6).

Users should note that `MATHPHA` is a moderately powerful tool, and thus open to confusion/abuse. Users are strongly urged to read & attempt to fully understand the deatils of the task. In particular, users should understand the implied units for any integers/reals which form part of the input expression, the details of the error propagation, and options available to calculate the exposure time written to the o/p file. Both of these issues are discussed in detail below. Any remaining confusion/uncertainties/strange results should be reported to the author.

At the current time, only a modest number of mathematical operations are possible (as listed in Section 2.1), though it is anticipated that they will serve the vast majority of users' needs. These will be added to as dictated by demand. In addition, due to the relative crudeness of the expression parser, there are number of rules which users must conform to when constructing their input expression (Section 2). Violation of these rules will result in the task (hopefully) stopping, (possibly) crashing, or (maybe even) producing incorrect results.

Only the the OGIP recognised PHA file formats described in OGIP/92-007

(see also Arnaud etal 1992) and its appendix provided in OGIP memo OGIP/92-007a are supported for both the input and output files.

2 RULES GOVERNING THE CONSTRUCTION OF THE INPUT EXPRESSION

The input expression is a character string (currently with a maximum length of 1024 characters) which forms a mathematical expression consisting of:

- PHA filenames
  Only the "obvious" restrictions are currently placed on filenames, namely that the path must be specified, and that the parser must not be able to interpret the path/filename as a mathematical operation, an integer/real, or some combination thereof (this currently poses a problem under unix & ultrix - see below). A specific extension of the input file containing the PHA dataset to be operated on can be specified by including the integer extension number in square ([extn#]) brackets after the filename. If such a specification is not used, then the entire PHA file will be searched for extensions likely to contain a PHA dataset. In such cases, if the input file is then found to consist of more than one PHA dataset, then the task will warn the user and stop.

  - At the current time, up to 10 filenames can be specified within the input expression (within the constraint of the total number of characters specified above). Within the above limitation on the number of filenames, a given i/p filename can feature any number of times within the input expression.

- Allowed mathematical operations
  The mathematical operators currently allowed are listed in Section 2.1. Any character which is not in this list, is not a curved bracket, and which cannot be interpreted as (part of) an integer/real, will be interpreted as a filename, or part thereof.

  - Users should adhere to standard mathematical conventions when constructing the input expression to ensure that operations are carried out in the desired order. It should be noted that in the case of a 'sub-expression' consisting of a series of multiplication and/or division operations on filenames and/or numerical constants (ie where the sub-expression does not contain either +, -, (or) characters), these operations will be carried out working from left-to-right. Thus the sub-expression 4/2*6 will result in a value of 12.
  - At the current time, the input expression can include any number of mathematical operations (within the constraint of the total number of characters specified above).
  - **NOTE** currently, a mathematical operator is **NOT** allowed to immediately follow a (character, or immediately preceed a) character in the input expression.

- Numerical constants (reals and/or integers)
  Numerical constants consist of a combination of ASCII characters in the range 0 to 9 along with an optional, single decimal point (.). For internal purposes, the parser interprets all integers as reals, thus the inclusion/exclusion of the decimal point for whole numbers is of no consequence.

  - It is **IMPERATIVE** that users remember that the units of any numerical constants added to or subtracted from a PHA dataset assumed to be in the units specified by the parameter units (the default of which is COUNTS; see Section 8), **IRRESPECTIVE** of whether the original PHA data themselves are stored as counts or counts per second.
  - At the current time, the input expression can include any number of numerical constants (within the constraint of the total number of characters specified above).

– **NOTE** at the current time numerals of the form `1E04` or `1E-04` are **NOT** supported (and will be interpreted as input filenames). This shortfall will be addressed in the near future.

- Matching pairs of curved brackets (parentheses)
  Curved brackets should be used in the standard mathematical mannor to denote the priority in which various calculations are performed. For safety, the task will stop if the number of open-brackets (`(`) does not equal the number of close-brackets(`)`). The only limitations currently in place are that neither the first nor last entry within a pair of brackets is a mathematical operator, in which case the task *SHOULD* stop.

  – At the current time, up to 10 pairs of brackets may be included within the input expression.

Within the constraint of the total number of characters specified above, any number of spaces can be added to the input expression.

2.1   Currently Supported Mathematical Operations

The following mathematical operations are currently supported:

- addition (`+`)

- subtraction (`-`)

- multiplication (`*`)

- division (`/`)

It should be noted that the parser will interpret all characters besides those listed above and the open/close curved brackets/parentheses characters (ie `(` and `)`) as (parts of) input PHA filenames.

3   CALCULATION & PROGATION OF ERRORS

There several options open to users regarding the calculation or propagation of errors, controlled by the parameters `errmeth` and `properr`.

The (hidden) parameter `errmeth` controls which prescription is used to calculate errors, should the task need to do so. The value of this keyword is therefore important if the errors are to be propagated (ie `properr='yes'`) and one or more of the i/p files contain the FITS keyword `POISERR = T` indicating Poissonian statistics should be used to calculate the statistical errors, or

if errors are not to be propagated (ie `properr='no'`) and the algebra is performed in counts/sec (ie `units='Rate'`).

Currently, the following prescriptions are available (where `N` is the number of counts observed in a given PHA channel):

- `errmeth = 'Gauss'`
  whereby the errors are calculated using $\texttt{error} = \sqrt{N}$.

- `errmeth = 'POISS-1'`
  (the default) whereby the algorithm of Gehrels (1986 ApJ, 303, 336) eqn 7 with S=1 is used ($\texttt{error} = 1.0 + \sqrt{N + 0.75}$). The value is statistically that of the (larger) +ve error of a Poissonian distribution, but within this task this value is assigned to both the +ve and -ve error on the counts in that channel. For small `N`, the errors created using this prescription is significantly **GREATER** (and hence more conservative) than both the true -ve error, and that obtained by simply using $\sqrt{N}$. Thus, this prescription is recommended unless a user fully understands the implications of using a different prescription. Both differences quickly reduce as one moves to larger `N` (and the Poissonian distribution becomes more symmetric/Gaussian).

- `errmeth = 'POISS-2'`
  whereby the algorithm of Gehrels (1986 ApJ, 303, 336) eqn 11, with S=1 is used ($\texttt{error} = \sqrt{N - 0.25}$). The value is statistically that of the (smaller) -ve error of a Poissonian distribution, but within this task this value is assigned to both the +ve and -ve error on the counts in that channel. This error prescription **UNDERESTIMATES** the true error for small `N`.

- `errmeth = 'POISS-3'`
  whereby the mean of the errors given by `POISS-1` and `POISS-2` above is used.

Caution is urged, particularly when using `errmeth = 'Gauss'`, as unexpected and/or misleading results can be produced.

The following tables enables direct comparisons to be made between these approximations:

The (hidden) parameter `properr` controls whether the errors are to be propagated during the algebra or (if `properr='no'`) whether the errors are simply calculated from the resultant PHA dataset. In the former case, the errors are propagated in the normal manner ($e_3 = \sqrt{[e_1^2 + e_2^2]}$). This is an approximation when in the Poissonian regime (ie for low `N`). Whilst it is true that the variances of the Poissonian distribution are combined in this normal way, confidence limit error bars are not simply related to the variance like they are for Gaussian statistics.

HOWEVER, these approximations work well for all but the smallest `N`, and is certainly superior to either assuming $\sqrt{N}$ errors, or neglecting errors altogther:

N	true 1-sigma Poisson errors		errors calcd using Gehrels approx		errors calcd using $\sqrt{\mathbb{N}}$	
0	+1.84	-0.00	+1.87	-0.00	+0.00	-0.00
1	+2.30	-0.83	+2.32	-0.67	+1.00	-1.00
2	+2.63	-1.92	+2.66	-1.33	+1.41	-1.41
3	+2.92	-1.63	+2.94	-1.66	+1.73	-1.73
4	+3.16	-1.91	+3.18	-1.94	+2.00	-2.00
5	+3.38	-2.16	+3.40	-2.18	+2.24	-2.24
10	+4.27	-3.11	+4.28	-3.12	+3.16	-3.16
50	+8.12	-7.05	+8.12	-7.05	+7.07	-7.07
100	+11.00	-9.98	+11.00	-9.99	+10.00	-10.00

- For example, adding two PHA datasets each with `N=5` counts in a given channel will give a value of `10`, and errors of:

  +4.24   -4.24 (using `errmeth='POISS-1'`) +3.08   -3.08 (using `errmeth='POISS-2'`)
  +3.95   -3.95 (using `errmeth='POISS-3'`) compared to:
  +4.27   -3.11 (statistically correct values)
  +3.16   -3.16 (propagating 'Gaussian' errors)

Severely misleading and/or incorrect results are possible if the errors are not propagated (ie if `properr=no`). Turning off error propagation is only reccomended when one is adding non-background-subtracted datasets, and one fully understands the risks.

4   CALCULATION OF THE EXPOSURE TIME IN THE O/P PHA DATASET

The value written as the exposure time in the o/p PHA dataset is indicated by a user-defined flag (`exposure`). This facility provides added flexibility to users, but with a risk of some confusion. It should be remembered however, that if desired, the exposure time given for a PHA dataset can always be reset using the `chkey` command within the ftools/heasarc task `GRPPHA`. The following options are currently available:

- `exposure = 'NULL'`
  where an exposure time of 1 second is written to the o/p dataset.

- `exposure =` *i/p filename*
  whereby the exposure time read from the specified i/p file is written to the o/p dataset.

- `exposure =` *real*
  where the entered value (assuming it can be parsed as a real), is written to the o/p dataset.

The value will be assumed to be in units of seconds. At the current time, numerals of the form `1E04` or `1E-04` are **NOT** supported (and will be interpreted as i/p filenames).

- `exposure = 'CALC'`
  where the exposure time is calculated from the i/p expression by sustituting each filename with its exposure time, and performing the specified calculation.

5 CALCULATION OF THE "AREA SCALING FACTOR" FOR THE O/P PHA DATASET

The area scaling factor is a numerical constant stored in the OGIP-mandatory FITS keyword `AREASCAL` in the `SPECTRUM` extension of a PHA file. Details of its use (and misuse) are too varied and complex to be fully described here, but most often it is used to renormalize the response matrix associated with that PHA file within `XSPEC`. As such the value of the `AREASCAL` keyword is often unity, but (particularly in the case of older response matrices) can have values of the order of the effective area (in cm^2) of the telescope/instrument.

The value written as the `AREASCAL` keyword in the o/p PHA dataset is determined by a hidden parameter, `areascal`. The following options are currently available:

- `areascal = 'NULL'`
  where an `AREASCAL` of 1.0 is written to the o/p dataset.

- `areascal = ` *i/p filename*
  whereby the `AREASCAL` read from the specified i/p file is written to the o/p dataset.

- `areascal = ` *real*
  where the entered value (assuming it can be parsed as a real), is written to the o/p dataset. At the current time, numerals of the form `1E04` or `1E-04` are **NOT** supported (and will be interpreted as i/p filenames).

- `areascal = '%'`
  (the default setting) where `MATHPHA` employs the following logic:

  - if the values of the AREASCAL keywords in the all the i/p files:
    * AGREE to within 1%
      this value (actually the straight mean) will be written to the o/p file.
    * DISAGREE by more than 1%
      the user is informed of the various values encounted in the i/p files, an `AREASCAL` of unity is written to the o/p file, and the user invited to determine and change the value of this keyword outside of `MATHPHA`.

In both cases, the user is informed of the value written to the o/p file. Essentially, if you dont see warning messages regarding `AREASCAL` during execution, don't worry about

it. If you do, and the values on the various i/p files vary dramatically and you dont understand why, then you should check whether the input expression was really that which you intended.

6 DETERMINATION OF THE AUXILIARY FILES FOR THE O/P PHA DATASET

There are a number of mandatory keywords required to be present in the header of extension containing the resultant PHA dataset in the o/p file. A number of these are crucial for the correct interpretation of the o/p dataset. The prime example is the value of the `EXPOSURE` keyword, and as described in Section 4, `MATHPHA` has a parameter `exposure` which controls how the value to be written to the o/p file is determined. Similarly `MATHPHA` compares the values of the `AREASCAL` keywords found in the i/p files, and attempts to determine the most appropriate value to be written to the o/p file (Section 5). `MATHPHA` also checks the values of the `TELESCOP`, `INSTRUME`, `DETNAM`, `FILTER` & `CHANTYPE` keywords. If there are discrepancies with any of these keywords, `MATHPHA` will warn the user and writes the appropriate null/default value to the o/p file.

However, there are a number of mandatory keywords associated with 'auxiliary' files, for which `MATHPHA` is in general unable to calculate 'appropriate' values (since it largely depends upon what the user is attempting to do via the input expression). These keywords are:

- `BACKFILE` - the name of the corresponding PHA file containing the appropriate background spectrum (if any)

- `BACKSCAL` - the numerical scaling which should be applied to the background spectrum (as specified by the `BACKFILE` keyword) before background-subtraction.

- `CORRFILE` - the name of the corresponding PHA file containing the appropriate correction spectrum (if any)

- `CORRSCAL` - the numerical scaling which should be applied to the correction spectrum (as specified by the `CORRFILE` keyword).

- `RESPFILE` - the name of the redistribution matrix (aka RMF) appropriate to the PHA dataset

- `ANCRFILE` - the name of the ancillary response dataset (aka ARF) appropriate to the PHA dataset, to be use in conjuction with the redistribution matrix given by the `RESPFILE` keyword.

(see also OGIP/92-007).

However, for convenience `MATHPHA` does provide users some opportunity to specify the values to be written to the o/p file via the hidden parameters `auxfiles`, `backfile`, `backscal`, `corrfile`,

`corrscal`, `arfile` and `rmfile` (see also Section 8). The internal logic for `MATHPHA` regarding these parameters is as follows:

1. If the `auxfiles`, `backfile`, `backscal`, `corrfile`, `corrscal`, `arfile` & `rmfile` parameters all have their default value ('NONE'), then the null/default values of the `BACKFILE`, `BACKSCAL`, `CORRFILE`, `CORRSCAL`, `RESPFILE` and `ANCRFILE` keywords will be written to the o/p file. The o/p file will thus have the same values for these keywords as obtained using versions of `MATHPHA` earlier than `v3.5.0`.

2. If the `auxfiles` parameter is set to one of the files in the i/p expression, then values of the `BACKFILE`, `BACKSCAL`, `CORRFILE`, `CORRSCAL`, `RESPFILE` and `ANCRFILE` keywords written to the o/p file will be taken (*ie* copied) from the specified i/p file (or set to their null/default values if there is a problem). It should be noted that setting the `auxfiles` parameter overrides any values specified via the `backfile`, `backscal`, `corrfile`, `corrscal`, `arfile` & `rmfile` parameters such that any values will be ignored.

3. If the `auxfiles` parameter is not set (has a value 'NONE'), then if any/all the parmeters `backfile`, `backscal`, `corrfile`, `corrscal`, `arfile` & `rmfile` are set to any of the files in the i/p expression, then values of the corresponding keyword(s) written to the o/p file will be taken (*ie* **COPIED**) from the specified i/p file (or set to their null/default values if there is a problem). At the current time, unfortuantely one **CANNOT** specify the name of (say) the response matrix to be written as the value of the `RESPFILE` keyword via the parameter `rmfile` – one can only specify the i/p PHA from which the `RESPFILE` is to copied to the o/p file.

7 WARNINGS ON USAGE

The parser is written in FORTRAN (!), and converts the user-supplied expression to "IMG" Reverse Polish (a little-known dialect thought to be understood by but one person on the planet). The parser attempts to identify the various elements of the input expression, and order the mathematical operations and the reading of PHA datasets appropriately. The parser attempts to work out from the last-entered (innermost) bracket-pair. The mathematical expression contained therewithin is evaluated and the answer stored. The parser then moves to the 2nd-to-last bracket-pair. The mathematical expression contained therewithin is evaluated (including the results already calculated, if appropriate), and the answer again stored. This process thus continues until the full expression has been evaluated, and this is the result which is finally written as the output PHA dataset.

7.1 Common Mistakes/Limitations

Users should keep in mind the following constraints/behaviour:

- At the current time, unix/ultrix users will to encounter severe problems if they attempt to specify the path to PHA datasets (not in the current directory): namely that the parser will interpret an input expression of the form:
  `expr = directory1/directory2/filename`
  to divide the PHA file called `'directory1'` by PHA file `'directory2'`, then divide the result by PHA file `'filename'`. This problem may be addressed in a future release of the task, however can be overcome in the meantime by copying the necessary input PHA file(s) to the current working directory.

- **BUG:** Due to the current lack of a unit-string parser, the units of the output file will always be either `'count'` or `'count/s'`. This is true even in cases such as `"expr='file1/file2'"`, or `"expr='file1*file2'"`. This is a bug which can lead to confusion, and a parser is being worked on.

- At the current time numerals of the form `1E04` or `1E-04` are **NOT** supported (and will be interpreted as input filenames).

- When a PHA dataset be either multiplied or divided by a numerical constant, the errors are calculated such as to keep the fractional error in the derived dataset equal to that in the i/p dataset.

The task will **STOP** (hopefully with an appropriate error message) if:

- The output file has an illegal name. For safety, under no circumstances can the o/p filename be the same as one of the i/p filenames. Under unix/ultrix, the task will also stop if a file of the same name already exists in the current directory, and the output file is NOT preceeded by `"!"` (see below).

- All the PHA files in the i/p expression do not have the same number of raw detector channels (as specified by the `DETCHANS` keyword in the i/p files).

- This task is not designed to replace pocket/on-line calculators, and (currently) will stop if the parser is unable to find something resembling a potential PHA filename within the entire input expression.

Users who encounter/suspect problems with the task are urged to contact the author, but also to experiment with the input expression (by increasing/moving pairs of brackets, and/or performing the required calculation with two or more passes through `MATHPHA`).

The author, USRA, the OGIP & NASA assume no responsibility for errors resulting from either bugs, or the misuse of this task. However, please report any comments/problems or bugs to Ian M George (`george@lheavx.gsfc.nasa.gov`).

8   INPUT PARAMETERS

- `expr [character string]`
  The i/p expression to be evaluated. This must represent a series of mathematical operations to be performed on (one or more) PHA datasets. The expression is allowed to contain PHA filenames (and optionally, an extension number), the mathematical operators listed above, integers, reals, and pairs of curved brackets (`(` & `)`). A number of other criteria, as specified above **MUST** also be satisfied.

- `units [character string]`
  A flag indicating the physical units in which the algebraic expression is to be evaluated (and the units in which the o/p file will be written). The allowed values are `'C'` (ie 'COUNTS' - the default), and `'R'` (ie 'RATE'), implying that the algebra will be performed in 'COUNTS' and 'COUNTS PER SECOND' space respectively. The algebra will be performed in this space irrespective of whether the input files contain data stored in counts or in counts per second (ie, if `units=C`, then i/p PHA histograms stored in counts/s will be converted to counts prior to any mathematical operations being performed). Similarly, this flag gives the implied units of any numerical constants within the input expression. A third option value is also allowed, `units=F` (ie 'FILE') whereby the algebra is performed in which ever units most of the files are stored in ('COUNTS' in the event of a tie). Extreme caution is urged using this final option.

- `outfil [character string]`
  The name of the o/p file to be written containing the derived PHA dataset. For safety, the name of the o/p file **CANNOT** be the same as one of the i/p files, and the task will stop if this is the case. Under unix/ultrix, an o/p filename identical to a file which already exists on disk is also considered illegal, and the task will stop. However, the existing file can be automatically removed, and the new file written if the o/p filename is preceded by `"!"` at the `outfil` prompt.

- `exposure [character string]`
  A flag indicating how the exposure time in the o/p PHA dataset is calculated/written. The following options are currently available: `NULL`, `CALC`, a real number (giving the exposure in seconds), or an i/p filename (see Section 4).

- `(areascal = '%')[character string]`
  A flag indicating how the `AREASCAL` keyword in the o/p PHA dataset is determined. The following options are currently available: `'NULL'`, `'%'`, a real number, or an i/p filename (see Section 5).

- `(properr='yes') [character string]`
  A flag whether the errors are to be propagated during the algebra (the default), or (if `properr='no'`) whether the errors are simply calculated from the resultant PHA dataset. The latter can be dangerous - see Section 3

- `(errmeth='POISS-1') [character string]`
  A flag indicating what error presciption is to be used should errors need to calculated by the task at any stage. The following values are currently allowed:

- errmeth = 'Gauss'
  whereby the errors are calculated using $\sqrt{\text{N}}$.

- errmeth = 'POISS-1'
  (the default) whereby the algorithm of Gehrels (1986 ApJ, 303, 336) eqn 7 (+ve error)
  is used.

- errmeth = 'POISS-2'
  whereby the algorithm of Gehrels (1986 ApJ, 303, 336) eqn 11 (-ve error) is used.

- errmeth = 'POISS-3'
  whereby $0.5 \times$ (POISS-1 + POISS-2) is used

Caution is urged, particularly when using errmeth = 'Gauss', as unexpected and/or
misleading results can be produced (see Section 3).

- ncomments [integer]
  The number of 70 character comment strings (up to a maximum of 4) to be added by the
  user. ncomments = 0 is allowed for users not wishing to add their own notes. The default
  value is 1.

- commentn [character string]
  The i/p parameters, including the expression, will be written as COMMENTS to the o/p file;
  these additional comments provide users the facility to add their own notes. Each string
  will be truncated beyond the 70th character.

- (auxfiles = 'NONE') [character string]
  A flag indicating whether the values of **ALL** the auxiliary file keywords (BACKFILE,
  BACKSCAL, CORRFILE, CORRSCAL, RESPFILE & ANCRFILE) written to the o/p file are to
  be copied from a user-specified i/p file. The default value ('NONE') indicates this is not
  the case, and will result in these keywords being written with their null/default values in
  the o/p file (see Section 6). It should be remembered that setting the auxfiles parameter
  overrides any values specified via the backfile, backscal, corrfile, corrscal, arfile
  & rmfile parameters (below) such that any values will be ignored.

- (backfile = 'NONE') [character string]
  A flag indicating how the BACKFILE keyword written in the o/p PHA dataset is determined.
  The following options are currently available:

  - backfile='NONE'
    The default value (NONE) for this keyword will be written in the o/p file

  - backfile='%'
    The task will check whether all the i/p PHA files have the same value of the BACKFILE
    keyword. If this is the case, then that value will be written to o/p file. Otherwise,
    the default value will be written to the o/p file.

  - backfile=*i/p filename*
    The value of the BACKFILE keyword from the specified i/p file will be copied to the
    o/p file.

In the latter two cases, any problems will be reported by the task, and the default value
written to the o/p file.

- (backscal = 'NONE') [character string]
  A flag indicating how the BACKSCAL keyword written in the o/p PHA dataset is determined. The following options are currently available:

  - backscal='NONE'
    The default value (1.0) for this keyword will be written in the o/p file

  - backscal='%'
    The task will check whether all the i/p PHA files have the same value of the BACKSCAL keyword. If this is the case (to within 1%), then that value will be written to o/p file. Otherwise, the default value will be written to the o/p file.

  - backscal=*i/p filename*
    The value of the BACKSCAL keyword from the specified i/p file will be copied to the o/p file.

  In the latter two cases, any problems will be reported by the task, and the default value written to the o/p file.

- (corrfile = 'NONE') [character string]
  A flag indicating how the CORRFILE keyword written in the o/p PHA dataset is determined. The following options are currently available:

  - corrfile='NONE'
    The default value (NONE) for this keyword will be written in the o/p file

  - corrfile='%'
    The task will check whether all the i/p PHA files have the same value of the CORRFILE keyword. If this is the case, then that value will be written to o/p file. Otherwise, the default value will be written to the o/p file.

  - corrfile=*i/p filename*
    The value of the CORRFILE keyword from the specified i/p file will be copied to the o/p file.

  In the latter two cases, any problems will be reported by the task, and the default value written to the o/p file.

- (corrscal = 'NONE') [character string]
  A flag indicating how the CORRSCAL keyword written in the o/p PHA dataset is determined. The following options are currently available:

  - corrscal='NONE'
    The default value (1.0) for this keyword will be written in the o/p file

  - corrfile='%'
    The task will check whether all the i/p PHA files have the same value of the CORRSCAL keyword. If this is the case (to within 1%), then that value will be written to o/p file. Otherwise, the default value will be written to the o/p file.

  - corrfile=*i/p filename*
    The value of the CORRSCAL keyword from the specified i/p file will be copied to the o/p file.

In the latter two cases, any problems will be reported by the task, and the default value written to the o/p file.

- `(arfile = 'NONE') [character string]`
  A flag indicating how the `ANCRFILE` keyword written in the o/p PHA dataset is determined. The following options are currently available:

  - `arfile='NONE'`
    The default value (`NONE`) for this keyword will be written in the o/p file
  - `arfile='%'`
    The task will check whether all the i/p PHA files have the same value of the `ANCRFILE` keyword. If this is the case, then that value will be written to o/p file. Otherwise, the default value will be written to the o/p file.
  - `arfile=`*i/p filename*
    The value of the `ANCRFILE` keyword from the specified i/p file will be copied to the o/p file.

  In the latter two cases, any problems will be reported by the task, and the default value written to the o/p file.

- `(rmfile = 'NONE') [character string]`
  A flag indicating how the `RESPFILE` keyword written in the o/p PHA dataset is determined. The following options are currently available:

  - `rmfile='NONE'`
    The default value (`NONE`) for this keyword will be written in the o/p file
  - `rmfile='%'`
    The task will check whether all the i/p PHA files have the same value of the `RESPFILE` keyword. If this is the case, then that value will be written to o/p file. Otherwise, the default value will be written to the o/p file.
  - `rmfile=`*i/p filename*
    The value of the `RESPFILE` keyword from the specified i/p file will be copied to the o/p file.

  In the latter two cases, any problems will be reported by the task, and the default value written to the o/p file.

- `(phaversn = '1.1.0') [character string]`
  The OGIP FITS format version for PHA datasets in which the o/p FITS file is to be written. The default is `PHAVERS1='1.1.0'` (also known as known as `PHAVERSN = '1992a'`), and is currently to only format allowed.

- `(divzero = -99) [real]`
  A real to be written into the o/p PHA dataset for channels which suffer a divide-by-zero during the evaluation of the i/p expression. The default value is `-99`.

- `(chatter = 9) [integer]`
  Flag to indicate how chatty the task is at execution. A value of `9` is the default, with lower/higher values producing quieter/verbose output respectively.

- (`clobber = false`) [`boolean`]
  Flag specifying whether or not a pre-existing file with the same name as that requested as the output file from this task will be overwritten. This parameter is required to be present by the FTOOLS group and NASA/GSFC and provides the same functionality as that described above whereby the o/p filename can be preceeded by `"!"` at the `outfil` prompt. It should be noted that the `"!"`*filename* syntax will overwrite any existing files irrespective of the value of the `clobber` parameter.

9 KNOWN BUGS

Unfortunately, the following bugs are known to exist in the current version of the task:

- Due to the current lack of a unit-string parser, the units of the output file will always be either `'count'` or `'count/s'`. This is true even in cases such as `"expr='file1/file2'"`, or `"expr='file1*file2'"`. This can lead to confusion, and a parser is being worked in.

10 WORKED EXAMPLES

Section incomplete

REFERENCES

Information regarding on-line versions of any of the following references with an OGIP Memo number (*ie* documents starting `OGIP/..` or `CAL/..`) can most easily be found via the WorldWide Web by following the links from the URL:
`/docs/heasarc/caldb/caldb_docs_index.html`

Most OGIP Calibration Memos of general community interest will eventually appear as articles in *Legacy*, but are also available on request from The Office of Guest Investigator Programs, Code 668, NASA/GSFC, Greenbelt, MD 20771, USA.

Arnaud, K.A., George, I.M., Tennant, A.F., 1992, *Legacy*, **2**, 65.
   (early, published version of OGIP/92-007)
George, I.M., Arnaud, K.A. 1992.
   (OGIP/92-007a

addendum to OGIP/92-007)
Gehrels, N., 1986, ApJ, 303, 336