

OGIP Calibration Memo CAL/SW/93-004

Summary of CALTOOLS Tasks

Ian M George, Ron S Zellar and Rehana Yusaf
Code 668,
NASA/GSFC,
Greenbelt,
MD 20771

Last Update: 2004 Jun 23

SUMMARY

This document summarizes all currently distributed tasks within the CALTOOLS sub-package of FTOOLS. The status of those CALTOOLS planned for the short-medium term future is also summarized.

This document is up-to-date for the FTOOLS version 3.4 Public Release and the FTOOLS in the development area at NASA/GSFC as of 1995 Oct 05.

Regarding those CALTOOLS planned, this draft is provisional.

LOG OF SIGNIFICANT CHANGES

Release Date	Sections Changed	Brief Notes
1994 Feb 10	All	FTOOLS Version 2.8 Public Release
1994 Mar 30	All	FTOOLS Version 2.9 Public Release
1994 Jun 01	All	FTOOLS Version 3.0 Public Release
1994 Oct 21	All	FTOOLS Version 3.1 Internal Release
1995 Jan 10	All	Made compatible with LaTeX2HTML software
1995 Jan 13	All	FTOOLS Version 3.2 Public Release
1995 Feb 22	All	FTOOLS Version 3.3 Public Release
1995 Oct 05	All	FTOOLS Version 3.4 Public Release
2004 Apr 01	All	Made compatible with tth
2004 Jun 23	All	FTOOLS Version 5.3.1 Public Release

Contents

1	INTRODUCTION	4
1.1	Overview	4
1.2	Summary of CALTOOLS Tasks	4
2	GENERAL CALIBRATION UTILITIES	5
2.1	Released General Utilities	5
2.1.1	ftools/caltools/addarf	5
2.1.2	ftools/caltools/addrmf	6
2.1.3	ftools/caltools/brcaldb	6
2.1.4	ftools/caltools/calcrpsf	7
2.1.5	ftools/caltools/dmprmf	7
2.1.6	ftools/caltools/gcorrmf	8
2.1.7	ftools/caltools/marfrmf	8
2.1.8	ftools/caltools/quzCIF	8
2.1.9	ftools/caltools/rbnrmf	8
2.1.10	ftools/caltools/rbnrpsf	9
3	FORMAT CONVERSION TASKS	11
3.1	Released Conversion Tasks	11
3.1.1	ftools/caltools/cmppha	11
3.1.2	ftools/caltools/col2img	11
3.1.3	ftools/caltools/rpsfqdp	11
3.1.4	ftools/caltools/rsp2rmf	11

<i>OGIP Calibration Memo CAL/SW/93-004 (CALTOOLS summary)</i>	4
---	---

3.1.5 ftools/caltools/st2rpsf	12
---	----

3.1.6 ftools/caltools/stw2pha	12
---	----

4 CALDB MAINTENANCE UTILITIES	13
--------------------------------------	-----------

4.1 Released Maintenance Utilities	13
--	----

4.1.1 ftools/caltools/chkcif	13
--	----

4.1.2 ftools/caltools/crcif	13
---------------------------------------	----

4.1.3 ftools/caltools/lstgood	13
---	----

4.1.4 ftools/caltools/mkcaldb	13
---	----

4.1.5 ftools/caltools/mkcaldir	14
--	----

4.1.6 ftools/caltools/mkcalinit	14
---	----

4.1.7 ftools/caltools/stcal	14
---------------------------------------	----

4.1.8 ftools/caltools/udcif	15
---------------------------------------	----

1 INTRODUCTION

1.1 Overview

CALTOOLS is a sub-package of the FTOOLS suite of software tasks maintained and distributed by the HEASARC within the Laboratory for High Energy Astrophysics at NASA/GSFC.

The CALTOOLS sub-package is dedicated to (mission independent) tasks relating to the creation and manipulation of calibration files, including tasks to create, maintain & access the HEASARC Calibration Database.

This document summarizes all currently distributed CALTOOLS tasks, and summarizes the status of those CALTOOLS planned for the short-medium term future.

1.2 Summary of CALTOOLS Tasks

A list of all general (multi-mission) CALTOOLS tasks currently distributed are listed in Table 1, and the status of those currently planned summarized in 2.

Table 1: Distributed CALTOOLS Tasks - General

Taskname	Author	Description
addarf	IMG	Adds two or more ancilliary response files (ARFs) with user-specified weightings.
addrmf	KAA	Adds together RMFs
brcaldb	RSZ	Browse the CALDB data holdings
calcrpsf	IMG	Multi-inst wrap to spawn inst-specific RPSF tasks
caldbflag	LB	switches flag status of calibration files in a CIF
caldbinfo	IMG	Checks out and reports upon the status of the CALDB available to a user
chkcif	MT	Checks entries in a CIF
cifcadd	LB	adds an extra column to a caldb.indx
cmppha	RY	Converts a Type II to Type I PHA file
col2img	IMG	Dumps a collimator resp dataset to an image
crcif	RSZ	Creates an empty Calibration Index File (CIF)
dmprmf	RY	Dumps a FITS RMF to screen or ASCII file
gcorrmf	RY	Remaps a RSP_MATRIX in channel space
lstgood	MT	Lists the Files with Quality = Good in a CIF
marfrmf	IMG	Multiples/Divides RSP_MATRIX by a SPECRESP array or constant
mkcaldb	RSZ	Create a Calibration Database
mkcaldir	RSZ	Create the Caldb directory structure
mkcalinit	RSZ	Create a caldbinit file
mudcif	LB	Enables Many Updates to a Caldb.Indx File
quzcif	RSZ	Quizes a Calibration Index File (CIF)
rbrmf	RY	Compresses an RMF file in channel-space
rbrpsf	RY	Rebins a radial psf (RPSF) dataset
RPS2EEF	BMWS	Converts a radial PSF (RPSF) dataset to a radial encircled energy function (REEF) dataset
rpsfqdp	RY	Dumps an (ASCII) QDP file from a RPSF file
rsp2rmf	IMG	Converts SF RSP file to a (FITS) RMF
st2rpsf	RY	Converts STWFITS o/p to a RPSF format file
stcal	RSZ	Stores one or more cal files in the Caldb
stw2pha	IMG	Convert STWFITS o/p to OGIP PHA FITS
udcif	RSZ	Updates a CIF for a new calibration dataset

2 GENERAL CALIBRATION UTILITIES

2.1 Released General Utilities

2.1.1 ftools/caltools/addarf

addarf sums input ARF files with specified weights, and outputs a single ARF. This task may be used when combining several spectral files having different ARFs but an identical RMF. For example, when combining energy spectra taken at different pointing positions with the same detector, and adding spectral files taken simultaneously with plural detectors having different ARFs but the same RMF.

Table 2: General CALTOOLS Tasks Under Development/Planned

Taskname	Author	Priority	Status	Description
calboon	IMG	1	0%	Switches datasets from being on- & off-line in a CIF
cifkys	IMG	1	50%	Writes or checks mandatory CIF keywords to FITS header
fakepha	IMG	1	10%	Folds fn of energy through RMF & makes PHA file
bldarf	IMG	2	0%	Multi-inst wrap to spawn inst-specific ARF tasks
bldrmf	IMG	2	0%	Multi-inst wrap to spawn inst-specific RMF tasks
img2psf	RY	2	0%	Calculates a rad rsf or EEf from a 2D psf image
psf2img	RY	2	0%	Calculates a 2D psf image from a rad rsf or EEf
rmf2rsp	IMG	2	10%	Converts FITS RMF to SF RSP
addcal	RSZ	3	10%	as for UDCIF, but with local additional features

Users are WARNED that the use of this task in inappropriate circumstances can (and most likely WILL) lead to extremely misleading results.

2.1.2 ftools/caltools/addrmf

`addrmf` adds together two or more RMFs with the provision so applying a user-defined weighting factor to each. The header and energy bounds extensions are copied from the first input RMF. Some checking is done to ensure that compatible RMFs are added however the author advises thought be applied before using this tool.

2.1.3 ftools/caltools/brcaldb

`brcaldb` is an interactive interface to the HEASARC's Calibration Database. It allows the user to repetatively filter a list of datasets based on their calibration characteristics and also provides the capability of copying datasets from the database to a user-specified directory.

Before using this task, the `CALDB` and `CALDBCONFIG` environment variables (logicals) must be set. See the `Caldb User's Guide` for details.

After executing the task, the user is shown a list of missions and instruments. The list displays those missions and instruments for which the Database contains data holdings. Immediately after this list is displayed the user is prompted to select those missions and instruments which s/he wishes to browse. The response to this prompt can contain multiple missions and instruments, but

each mission/instrument set should be separated by a comma.

After the appropriate mission and instrument data are loaded into memory, the user is free to enter any valid `brcaldb` command. The commands are described in the command-line help. From within `brcaldb`, type `help` for more information.

2.1.4 `ftools/caltools/calcrpsf`

`calcrpsf` This task is a multi-task 'wrapper' to run several `ftools` associated with Radial (1-dimensional) Point Spread Functions (RPSFs). Currently, the following tasks are supported:

- `ftools/caltools/st2rpsf` (see Section 3.1.5)
 - which reads i/p data from a FITS file produced by `IRAF/stsdas/fitsio/stwfits` assumed to contain a 1-dimensional radial profile of an image, and writes an o/p RPSF dataset in OGIP-standard format.
- `ftools/caltools/rbnrpsf` (see Section 2.1.10)
 - which rebins an RPSF dataset such that a user-defined number of 'source' counts are contained within each bin, and writes a new file containing the results.
- The following theoretical RPSF generators:
 - `ftools/rosat/pcrpsf` - for the ROSAT PSPC
 - `ftools/rosat/hrirpsf` - for the ROSAT HRI
- `ftools/caltools/rpsfqdp` (see Section 3.1.3)
 - which reads an observed RPSF dataset and writes the data to an ASCII file along with QDP commands to enable QDP/PLT to be used directly on the o/p file. In addition, a theoretical/predicted RPSF dataset can be read and written such that QDP/PLT will overlay it on the same curve.

One, some or all (appropriate) tasks can be spawned automatically from within `calcrpsf`, as dictated by the input parameters. More detailed help on the individual tasks can of course be obtained below, or by using the command `fhelptask`. A more detailed description of `calcrpsf` can be obtained from the memo CAL/SW/93-011 (available online in pdf and HTML versions).

2.1.5 `ftools/caltools/dmprmf`

`dmprmf` displays the contents of an RMF file (containing the detector response matrix) either to the terminal, or alternatively to an ASCII file if the user enters an output filename. Both EBOUNDS and RMF extensions can be displayed with user-defined channel or energy ranges (respectively).

2.1.6 **ftools/caltools/gcorrnmf**

gcorrnmf enables a detector response matrix to be shifted in channel-space. As such it should be used with **extreme** care.

2.1.7 **ftools/caltools/marfrmf**

marfrmf A detector redistribution matrix (aka response matrix) basically consists of a 2-dimensional array (detector channel vs incident photon energy) giving the probability that an incident photon of a given energy will be detected in a given detector channel. An ancillary response dataset is a 1-dimensional array as a function of incident photon energy (containing, for example, the effective area of the optics, corrections for the spatial point spread function etc). Provided the input files are in an OGIP-approved FITS format, this task provides the facility to multiply the redistribution array for each channel (along the incident energy dimension) by the ancillary response array. The output is thus a new FITS 'RMF' file containing the new redistribution matrix.

2.1.8 **ftools/caltools/quzCIF**

quzCIF This task is the primary interface software for the Calibration Database. It will return the location of a calibration dataset for a particular mission and instrument by specifying the type of dataset requested (codename), the date for which the dataset is to be valid, and the time for which the dataset is to be valid. If a dataset which meets these parameters are found, **quzCIF** will return the location of the data via the parameters filename and extension number.

Optionally, the **prscreen** parameter can be used to print this information to the screen. If **prscreen** = yes, the full path and filename will be printed to the screen along with the extension number. These two values will be separated by a space, and will appear on the same line.

If no datasets are found, then the message "The requested dataset is not in the Calibration Database" is displayed.

The value "NOW" can be entered for the date and time parameters. This value will cause **quzCIF** to use the current system date and time for the date and time parameters.

Before using this task, the Calibration Database Environment must be setup and enabled. See OGIP Memo CAL/GEN/92-008 for more details.

2.1.9 **ftools/caltools/rbnrmf**

rbnrmf compresses a FITS RMF matrix (redistribution matrix file in an OGIP-standard format)

in channel-space to give a user-defined number of resultant channels. The output is a new file containing the revised (compressed) RMF extension, and the corresponding "EBOUNDS" extension containing the nominal energies of the revised set of channel boundaries.

Caution is urged before `rbrnrmf` is used. The primary use of the task is likely to be to compress RMF datasets in channel-space such that the resultant number of channels matches that in a given PHA dataset. Hence downstream spectral analysis packages can be used to unambiguously map the channels in the PHA spectrum and Response matrix. The mode (or 'type') of compression performed is specified via the user-defined "cmpmode" parameter.

In order to minimise the disk-space requirements, the RMF matrix is stored in compressed form (CAL/GEN/92-002), that is all matrix elements below a given threshold (specified by `LO_THRES` keyword) are not stored. Unfortunately the `LO_THRES` keyword is not always used correctly, that is elements below this value are stored. When `rbrnrmf` is used on such a matrix, the values below `LO_THRES` will NOT be stored. In order to overcome this, users may use the `ftool/futil` `FPARKEY` to change the `LO_THRES` keyword value in the input RMF file.

The non-linear compression modes available for use with ASCA SIS data should be used with extreme care. These `rbrnrmf` modes are only required if users wish to perform spectral analysis on a PHA dataset taken in one on-board datamode in conjuncture with a detector response designed for use with PHA data obtained in the other datamode. Users are warned that it is not always immediately obvious if a PHA file using one channel-numbering scheme is used with a matrix using the other (though sharp discontinuities are a dead giveaway). Unfortunately, at the present time it is not possible to provide software checks for such errors, and it is the responsibility of the user to determine & pay attention to the channel-numbering scheme used to construct the PHA & RMF files.

2.1.10 `ftools/caltools/rbrnrpsf`

`rbrnrpsf` reads an i/p FITS datafile containing a radial point spread function dataset in OGIP-standard format, and rebins the dataset such that each new bin contains a user-supplied minimum number of counts. The result is written in the form of a BINTABLE (with `extname = OBS RPSF`) in the output file (again in OGIP standard format). Details of the o/p file formats supported can be found in the OGIP Calibration Memo CAL/GEN/92-020 available via the OGIP anonymous ftp account on `heasarc.gsfc.nasa.gov`.

The number of ('source') counts in each radial bin is calculated after the subtraction of an appropriate background. By default this background is obtained from the `BACKGRND` keyword within the i/p file (and may be zero). However users may override this value by entering their own (see below).

If a radial profile has been background subtracted, with an overestimate for the background rate, negative counts can be present in the input datafile. It is important to note that when `rbrnrpsf` performs the rebinning, these negative values are (negatively) included within the summation to

determine whether the requested number of minimum counts are in the new bin. This often results in wide output bins.

In order to obtain a good resolution extract the radial profile with many steps, for example if the IRAF task IMCNTS is used to perform the extraction from 0 to 10 arcminutes, use at least 80 steps.

3 FORMAT CONVERSION TASKS

3.1 Released Conversion Tasks

3.1.1 `ftools/caltools/cmppha`

`cmppha` This task converts an OGIP standard TYPE II pha file to a TYPE I pha file. A TYPE II pha file contains one extension with one or more spectra stored as rows. A TYPE I pha file contains only one spectrum in an extension. This task has two modes, Squish, and Expand. Squish compresses (adds) all the TYPE I spectra to form one spectrum, and writes it to the output file. Expand writes an extension for each spectrum. The resultant output file can then be used by other `ftools` (eg `GRPPHA`) that expect a TYPE I pha file as input.

3.1.2 `ftools/caltools/col2img`

In general, the collimator response of a collimated instrument is a function of spatial position and energy (where the latter arises due to photon scattering and/or transmission by the collimator). As a result, in the general case the OGIP format for such datasets consists of a single-row BINTABLE extension with a column containing the 3-dimensional (X,Y,E) array of collimator responses, along with other columns giving the spatial and energy grids used. Further information on the detailed format can be found in OGIP Calibration Memos CAL/GEN/92-022.

`col2img` is primarily for diagnostic/visualization purposes. The task reads a 3-D (X,Y,E) collimator response dataset, compresses the energy dimension (optionally over a user-defined energy range), to produce an output file containing an (X vs Y) image in the Primary Array (optionally rebinned/remapped by user-defined factors).

3.1.3 `ftools/caltools/rpsfqdp`

`rpsfqdp` reads i/p data from a FITS OGIP standard format file, containing an extension with a radial profile (extname = OBS RPSF), and an optional extension containing a theoretical/predicted model of the data (extname = THEO PSF). It subsequently writes an o/p file in ascii qdp format, so that QDP/PLT can be used directly on this file.

3.1.4 `ftools/caltools/rsp2rmf`

`rsp2rmf` reads i/p data from an old-stle (SF) format response matrix (used by XSPEC versions <8.2) and writes an o/p FITS data file in OGIP standard format. Currently only OGIP standard RMFVERS='1.1.0' (formerly '1992a') is defined & supported.

3.1.5 ftools/caltools/st2rpsf

`st2rpsf` reads i/p data from a FITS file produced by `IRAF/stsdas/fitsio/stwfits` assumed to contain a 1-dimensional radial profile of a image, and writes an o/p FITS data file in OGIP standard format for radial profiles. Currently only OGIP standard RPSFVER=1992a is defined & supported. The data is written in the form of a BINTABLE in the first extension of the o/p file (with EXTNAME = OBS RPSF).

3.1.6 ftools/caltools/stw2pha

`stw2pha` reads i/p data from a FITS file produced by `IRAF/stsdas/fitsio/stwfits` assumed to contain a 1-dimensional PHA histogram, and writes an o/p FITS data file in OGIP standard format for such a dataset. This involves writing both general & instrument-specific information into the FITS file.

4 CALDB MAINTENANCE UTILITIES

4.1 Released Maintenance Utilities

4.1.1 `ftools/caltools/chkcif`

`chkcif`: This task will examine the entries in a specified Calibration Index File. The CIF can be named on the command line otherwise the user will be prompted for the desired mission and instrument. An attempt will be made to open each calibration file listed in the CIF and to advance to the proper extension. Errors in opening/advancing the files are reported to either a logfile or to the standard output as specified in the outfile parameter.

4.1.2 `ftools/caltools/crcif`

`crcif`: This task will create a Calibration Index File with no rows. By default the CIF's name will be `caldb.indx`, however this can be changed by using the filename parameter on the command line.

4.1.3 `ftools/caltools/1stgood`

`1stgood`: This task will run through a specified Calibration Index File and list (to outfile) the full pathname of each calibration file which is flagged as ONLINE and of good quality. The CIF can be named on the command line, otherwise the user will be prompted for the desired mission and instrument.

4.1.4 `ftools/caltools/mkcaldb`

`mkcaldb`: When executed this program will perform the necessary tasks required to construct a Calibration Database.

First, a `caldbinit` file is created. This is the file that contains the system specific commands used to create the Caldb environment variables or logicals. (See the Caldb help page for more info about how this file should be used.) On UNIX systems the `caldbinit` file is created in the subdirectory `software/tools` and is called `caldbinit.unix`. On VMS systems the `caldbinit` file is created in the subdirectory `[.software.tools]` and is called `caldbinit.vms`.

Next, the Caldb directory structure is created along with the Calibration Index Files. This step is performed by reading the `caldbconfig` file, which this task expects to find in the `software/tools` or `[.software.tools]` subdirectory. It also expects this file to be called `caldb.config`. For more

information about the format and substance of this file, see the `caldb.config` header comments or the Caldb User's Guide.

Finally, `mkcaldb` will move any calibration files specified by the `files` parameter to their appropriate storage directories.

Should any errors occur during the execution of this task, the user will be told, but the task will continue to execute in an effort to do as much as possible toward the creation of a Caldb.

4.1.5 `ftools/caltools/mkcaldir`

`mkcaldir`: This task creates `bcf` and `cpf` storage subdirectories beneath each directory specified in the `caldbconfig` file. These subdirectories store "Basic Calibration Files" and "Calibration Product Files", respectively. If the directory specified in the `caldbconfig` file does not exist, it will be created. Optionally, empty Calibration Index Files also specified in the `caldbconfig` file can be created by this task.

The path to the `caldbconfig` file is obtained from the `CALDBCONFIG` environment variable or logical. The user may also need to set the `CALDB` environment variable or logical before executing this task. See the Caldb help page and Caldb User's Guide for more information on how to set these system variables.

4.1.6 `ftools/caltools/mkcalinit`

`mkcalinit`: This task will create a `caldbinit` file which contains the system specific commands necessary to set the `CALDB` and `CALDBCONFIG` environment variables or logicals. On UNIX systems the `caldbinit` file is created in the subdirectory `software/tools` and is called `caldbinit.unix`. On VMS systems the `caldbinit` file is created in the subdirectory `[.software.tools]` and is called `caldbinit.vms`. Caldb users can execute this file to initialize their session for Caldb access. See the Caldb help page for additional information on how to use the `caldbinit` file.

4.1.7 `ftools/caltools/stcal`

`stcal`: This task will open each calibration file specified by the `files` parameter and search for the keywords `TELESCOP`, `INSTRUME` and `CCLS0001`. When all three keywords are found, the `caldbconfig` file is consulted to determine the directory to which the calibration file should be moved. If the `CCLS0001` keyword value is "BCF" then the calibration file will be moved to the `bcf` storage directory of the directory specified in the `caldbconfig` file. If the `CCLS0001` keyword value is "CPF" then the calibration file will be moved to the `cpf` storage directory.

4.1.8 **ftools/caltools/udcif**

udcif: This task will search a FITS file for required Calibration Database keywords. When a complete set is found, a new entry is created in the Calibration Index File.

Should the new entry contain information which duplicates a previous entry and if the new entry is being entered with quality = 0, then the user is prompted to reconcile the duplication.