OGIP Calibration Memo CAL/GEN/92-003

Basic Calibration File (BCF) & Calibration Product File (CPF) Guidelines

Ian M George

& Ron S Zellar

Code 668,
NASA/GSFC,
Greenbelt, MD20771

Version: 1995 Jul 11

SUMMARY

The basic guidelines for the creation of multi-mission HEASARC-standard FITS formats for Basic Calibration Files (BCFs) and Calibration Product Files (CPFs) are described.

Intended audience: authors of HEASARC calibration data, and HEASARC programmers

Forward

This document is intended to provide a general overview of the calibration file formats used within the HEASARC calibration database. Every attempt has been made to present the information in a clear, and (moderately) concise manner, and copious use of cross-referencing is made, although reference to other Calibration Memos *etc* is occasionally required. The document is intended to be used as a manual, rather than read as a novel.

At the current time, this document is very much 'live' in the sense that it is being continually updated, and not guarenteed complete. Furthermore, a number of the rules/conventions described within should be considered unstable, and may change a short notice in the near future. Authors of calibration datasets are therefore urged to contact the authors to ensure they have the latest copy.

All questions, comments and suggestions should be directed to:

caldbhelp@olegacy.gsfc.nasa.gov.

LOG OF SIGNIFICANT CHANGES

Release Date	**Sections Changed**	**Brief Notes**
1992 Jun 06	First Draft	
1993 Jan 25	All	Special issue for XTE GOF
Apr 25	Coordinate frames	Reviewed & updated
1993 Oct 04	All	Detailed Formats split off
1994 Jan 05	The CBD*nxxxx* String	New syntax introduced
Jan 06	Coordinate frames	LINX,LINY changed to DETX,DETY
1994 Aug 09	Section 4.1	Added *i*CTYP*nnn* convention
1995 Jan 27	All	Made compatible with `LaTeX2HTML` software
1995 Jul 11	11	Added `XMA_DCOS` Coordinate system
	4.2 & 4.3	Revised syntax
2004 Apr 01	All	made compatible with tth
2007 Mar 21	title	changed title, minor typos fixed

Contents

1 INTRODUCTION 1

2 CALIBRATION INDEX FILE REQUIREMENTS 3

2.1 Keywords Required for CIFs 3

2.2 The parameter-space limitations of datasets (the CBD*nxxxx* keyword) 4

2.2.1 Syntax of the CBD*nxxxx* boundary keywords 4

2.2.2 Allowed formats of *expr* 5

3 OVERVIEW OF CALIBRATION DATA TYPES 7

4 USE OF MULTIDIMENSIONAL DATASETS 7

4.1 Ordering of Multidimensional Datasets 8

4.2 The 'axis labels' for multidimensional columns 8

4.3 Cross-referencing 'axis grid' with columns 9

4.4 Number of elements in 1-d arrays 10

4.5 Specifying Columns as Keywords 10

4.6 Unnecessary Columns & Keywords 10

5 THE USE OF VIRTUAL CALIBRATION FILES 10

5.1 Mandatory Keywords for Virtual Calibration Files 11

6 PHYSICAL UNITS OF DATASETS 11

7 PHOTON ENERGY GRIDS 12

8 THE SPECIFICATION OF TIMES 12

9 ERRORS ON DATASETS **13**

10 HDUCLASn & HDUVERSn KEYWORDS **14**

11 COORDINATE FRAMES **15**

11.1 Allowed CSYSNAME values 15

11.2 Pixel Definitions 17

11.3 Naming Convention for Columns Describing Spatial Coordinates 18

1 INTRODUCTION

In line with IAU and NASA policy, all files within the HEASARC make use of the Flexible Image Transport System (FITS; *eg* see Wells *et al.* 1981, Griesen & Harten 1981). The files make use of (and conform to) all the recent enhancements of the original FITS formats. Specifically, wide use is made of 'extensions' (Grosbol *et al.* 1988), 'ASCII tables' (Harten *et al.* 1988), and, in particular, 'binary tables' (Cotton & Tody 1992) FITS enhancements. The BCFs & CPFs provide no exception to this.

Within the calibration database (caldb), calibration files have been classified into three types:

- Primary calibration Files (PCFs), containing the 'raw' results of ground and in-orbit calibration observations (and hence not of direct interest to the general user);
- Basic Calibration Files (BCFs), representing the lowest level calibration information for a given instrument which is supplied to general users;
- Calibration Product Files (CPFs), containing calibration information customized for a particular analysis task or scientific dataset.

BCFs are thus constructed from the PCFs (along with any necessary algorithms and theoretical considerations), and are used and combined during the construction CPFs. BCFs are, of course, also used directly by (scientific data) analysis packages.

Due to various logistic reasons, and as an aide to clarity, the HEASARC calibration data is stored in a relatively large number of small BCFs & CPFs. Broadly speaking, each file is dedicated to a single aspect of calibration, and consist of a single FITS extension containing a single calibration data array. In some cases, however, a few closely related aspects of the same calibration (*ie* closely related data arrays) may be combined within a single file in order to save disk space. In the event of a calibration update, the new calibration information will be written as a new BCF/CPF, which (wherever possible) mirrors the format corresponding old BCF/CPF. The old BCF/CPF, however, will of course be retained within the archive and accessible to users and Stage 2 Cal s/w.

The FITS header of each file contains all neccessary details concerning the contents, orgin, time-tagging for use, and (to a limited extent) information on the use of the data within. The is primarily achieved via Calibration Index Files (see Section 2), along with a number of mandatory FITS keywords. For convenience, Table 1 lists all mandatory keywords for the files (beyond the required standard FITS keywords) cross-referenced to the relevant section of this document. Detailed descriptions of the origin, contents and use of all BCF & CPFs will be given in the relevant instrument Calibration Guide.

This memo is concerned with the general guidelines for constructing a new BCF or CPF format for use within the Within the HEASARC caldb. This document does **NOT** describe the detailed format of the individual calibration files. However references to such formats can be obtained from CAL/GEN/92-011 (George, Zellar & Pence 1992).

Table 1: Summary of Mandatory keywords required in BCFs & CPFs

Keyword Name	**Description**	see Section
Mandatory keywords always required		
`CCLS`*xxxx*	HEASARC-class of calibration file	2.1
`CCNM`*xxxx*	extension codename	2.1
`CDES`*xxxx*	descriptive string	2.1
`CDTP`*xxxx*	datatype code	5 & 2.1
`CVSD`*xxxx*	validity start date	2.1
`CVST`*xxxx*	validity start time	2.1
`INSTRUME`	instrument name	2.1
`TELESCOP`	satellite/mission name	2.1
Keywords mandatory required under certain circumstances		
i`CTYP`*nnn*	The 'axis-labels' of *n*-d array	4.2
`CBD`*nxxxx*	array describing parameter limitations of the dataset	2.1 & 2.2
`CREF`*nnn*	The axis–to-column cross-referencing of *n*-d array	4.3
`CSYSNAME`	spatial coordinate system in use	11
`CTASKLAN`	language in which a standalone task was written	5.1
`CTASKLOC`	disk location of executable of a standalone task	5.1
`CTASKNAM`	name of standalone task	5.1
`DETNAM`	detector name (if value of `INSTRUME` insufficient)	2.1
`FILTER`	filter in use	2.1
`TDIM`*nnn*	Number of elements & Ordering of *n*-d array	4.1

where:
xxxx is a reference number in the format *xxxx* = `0001`, `0002`, ... *etc*
nnn is the column number in the format *nnn* = `1`, `2`, `3`, ... *etc*
n is a reference integer between `1` and `9` (see Section 2.2).

2 CALIBRATION INDEX FILE REQUIREMENTS

In order to facilitate s/w and user identification/access of the numerous calibration datasets within the Within the HEASARC caldb, the location, contents and quality of all datasets will be contained within Calibration Index Files (CIFs). A detailed description of the contents, format and operation of CIFs is given in CAL/GEN/92-008 (George, Pence & Zeller 1992). A number of mandatory keywords are therefore required to be present within the extension header of each and every calibration dataset contained within the HEASARC caldb. It is worth stressing that it is intended that the CIFs will provide the main (possibly only) link between s/w and the calibration datasets. Most of the tasks which have been written to create and update the CIFs will ignore datasets not containing these keywords. Such datasets will therefore not be represented in the CIF, and hence inaccessible to downstream s/w.

2.1 Keywords Required for CIFs

The following keywords are mandatory if a calibration dataset is ever to form an entry in a CIF:

- `TELESCOP` - the name of the satellite/mission.
- `INSTRUME` - the name of the instrument.
- `CCLS`*xxxx* - the HEASARC-class of this calibration file.
  Of relevance here:
  - `CCLS`*xxxx* = `'BCF'`, for Basic Calibration Files
  - `CCLS`*xxxx* = `'CPF'`, for Calibration Product Files
- `CDTP`*xxxx* - code describing whether the 'calibration dataset' consists of real or 'vitual' data (see Sections 3 & 5)
- `CCNM`*xxxx* - codename of the extension to be used within CIF to describe the contents (for downstream s/w).
- `CVSD`*xxxx* - the UTC date (in `dd/mm/yy` format) when this calibration data should first be used.
- `CVST`*xxxx* - the UTC time (in `hh:mm:ss` format) on the day `CVSD`*xxxx* when this calibration data should first be used.
- `CDES`*xxxx* - a string giving a brief descriptive summary of this dataset.

and the following keywords are mandatory under certain circumstances:

- `DETNAM` - the name of the specific detector to which the calibration applies should the value of the `INSTRUME` keyword be insufficient.

- `FILTER` - (if applicable) the name of the (moveable) filter in position for the calibration dataset. This keyword is not required for instruments without a moveable filter.
- `CBD`*nxxxx* - (see Section 2.2) an array of strings (with *n* arbitrary integers between `1` & `9`) giving the parameter limitations of the dataset (*eg* energy range range, off-axis angle *etc*) used within the CIF to further describe the contents for downstream s/w (in association with the value of the `CCNM`*xxxx* keyword).

where *xxxx* is a number of the form `0001`, `0002`, `0003` *etc* required to identify the respective set of keywords associated with each dataset should two or more datasets be included within in single extension. In the vast majority of cases, only one calibration dataset is stored in a given extension (indeed, this is strongly recommended), thus *xxxx* = `0001` can be used.

The appropriate values of those keywords listed above which are not immediately obvious are given in the appropriate memo describing the format of the format of each calibration datatype. A full set of allowed values for each keyword can be found in CAL/GEN/92-011 (George, Zellar & Pence 1992).

2.2 The parameter-space limitations of datasets (the CBD*nxxxx* keyword)

The calibration boundary keywords provides a means of specifying the limitations or parameter boundaries of a calibration dataset (*eg* the energy range, range of spatial coordinates, range of temperatures, range of HV settings *etc* over which the dataset is valid) which the author of the dataset would like to indicate to downstream s/w. The calibration boundary keywords are named `CBD`*nxxxx* where *xxxx* is the calibration dataset whithin that extension (as described above), and *n* is an integer index in the range `1` − `9` specifying the boundary reference number[1]. Thus the limits on each calibration dataset within an extension can be denoted via the keywords `CBD1`*xxxx*, `CBD2`*xxxx*, `CBD3`*xxxx*, ..., `CBD9`*xxxx*. The ordering of the various strings is not crucial (*ie* which parameter limitations is specified by `CDB1`*xxxx*, which by `CDB2`*xxxx etc* is not crucial), although the values of *n* within the `CBD`*nxxxx* keywords **must be sequential starting at 1**. However, when checking for any limitations on a given parameter the (CIF) access s/w will first check the string stored in `CDB1`*xxxx*, then `CDB2`*xxxx etc*, thus it an advantage to store the most important limitations (from the point of view of downstream s/w) first.

2.2.1 Syntax of the CBD*nxxxx* boundary keywords

The value of each `CBD`*nxxxx* keyword is a character string which refers to a different dimension of parameter space, and has the following format:

`CBD`*nxxxx* = '*expr(VALDES1, VALDES2,..., VALDESj)units*'

[1] It is anticipated that a maximum of 9 will be easily sufficient for all calibration datasets, though an extension making *n* a hexadecimal number is possible if this is not the case (however this is not implemented at the time of writing).

where *expr* is a special character string describing the boundary parameter, *VALDESj* is the j^{th} value descriptor specifying the set of values for which the parameter is valid, and *units* is the physical units of *VALDESj*. The allowed values of *expr* are listed in Section 2.2.2, and the allowed values of the *units* string are as for the `TUNITS`*nnn* keyword and summarized in OGIP/93-001 (George & Angelini 1994).

Each of the value descriptors, *VALDESj*, can have three possible forms, any of which can be included/combined in the same `CBD`*nxxxx* keyword value:

1. *VALDESj* = *m.n*
   respresenting a single real number, *m.n*, for which the boundary parameter is valid. Only a single integer *m* need to specified in the case of an integer boundary value.
2. *VALDESj* = *minval-maxval*
   representing a range of continuous values between *minval* and *maxval* (where *minval* and *maxval* both have the form *m.n* given above) for which the boundary parameter is valid.
3. *VALDESj* = *cstr*
   representing a single character string, *cstr* for which the (character) boundary parameter is valid. In this syntax, double quotes (") can be included as the first and last characters of *cstr* to force a value to be interpreted as a string. The double quotes are **mandatory** if *cstr* would otherwise be interpreted as either a single integer/real or as a ranges of integers/reals.

Examples of these formats are given in CAL/GEN/92-011 (George, Zellar & Pence 1992).

2.2.2 Allowed formats of *expr*

Currently only the most simple type of parameter expression is supported, namely a format in which the *expr* string is simply the name of a parameter, *pname*, denoting that the calibration dataset is valid for parameter *pname* values between *min* and *max* (in units given by *units*). The allowed values of the *pname* string are as for the standard column/keyword names listed in CAL/GEN/92-011 (George, Zellar & Pence 1992

Example
A calibration dataset, which was the only such dataset within the extension (hence had *xxxx*=`0001`), and which was valid for photon energies in the range 0.501–2.0 keV, off-axis angles in the range 0–54.2 arcmin, and all azimuthal angles (0–360°) would have

```
CBD10001 = 'ENERG(0.501-2)keV'
CBD20001 = 'THETA(0-54.2)arcmin'
CBD30001 = 'PHI(0-360)deg'
```

A calibration dataset, which was the only such dataset within the extension (hence had *xxxx*=`0001`), and which was valid for photon energies in the range 1 eV – 10 MeV, an off-axis angle 5.4 arcmin (only), and azimuthal angles 0–90° and 180°–270° (but nowhere else) would have

```
CBD10001 = 'ENERG(1-10000000)eV'
```

```
CBD20001 = 'THETA(5.4)arcmin'
CBD30001 = 'PHI(0-90,180-270)deg'
```

The number of parameter-space limitations (*ie* `CBD`*nxxxx* keywords) required for a given calibration dataset depends upon the dataset itself (obviously), the characteristics of the specific instrument to which it refers, and the likelihood that other ($Qual = 0$) datasets with the same `CCNM`*xxxx* codename will ever exist in the archive at any time.

The following two detailed examples should help illustrate this point:
Example 1:
Consider an imaging instrument for which one requires to store a series of point-spread-function *psf* calibration datasets for various off-axis positions in the form of radial-profiles. However, it is known (or suspected) that the *psf* is a function of energy, yet the energy dependency has not (yet) been adaquately parameterized such that the datasets can be stored as a virtual calibration file and standalone s/w task). One therefore wishes to store radial profiles appropriate for several 'standard' energy ranges (*eg* in the 3 bands 0–1 keV, 1–2 keV & 3–3.5 keV) in separate files. The HEASARC codename for a *psf* in the form of a radial profile is simply `CCNM`*xxxx* = `'RPSF'` (CAL/GEN/92-020, George & Yusaf 1992a), with no reference to the energy band. Thus, there would be 3 datasets (files) indexed within the CIF with an identical `CCNM`*xxxx* value, and CIF access s/w could not be able to distinguish between them unless the files also contained `CBD`*nxxxx* keywords with values
`ENERG(0-1)keV`
`ENERG(1-2)keV`
`ENERG(3-5)keV`
respectively. The presence of such keywords within the datasets results in identical entries in the `CAL_CDB` column of the CIF. This column is read and parsed by the CIF access tasks/routine (`QUZCIF/QZCIF` respectively; see (CAL/SW/92-017, Zellar & George 1993), and thus should downstream s/w request a radial profile applicable for (say) 0.53 keV incident photons, `QUZCIF` will return the location, filename *etc* of the 1st dataset. Similiarly if a radial profile applicable to an incident energy of 2.5 keV is requested by downstream s/w, `QUZCIF` will report that no such calibration dataset is currently available.
Example 2:
Continuing from the above example, consider now that it is suspected that the *psf* may also be a function of detector temperature. If the above datasets (say all taken with a detector temperature of 273 K) also contained `CBD`*nxxxx* keywords with values
`TEMP(273)K`
then if the suspected temperature dependency is later indeed found to be present, `QUZCIF` knows immediately that all 3 datasets are inappropriate for observational data taken with a detector temperature of 200 K.

Clearly the h/w and GOF teams with have the best idea as to which parameter limitations are likely to be necessary for a given dataset, and thus the specification of of the necessary `CBD`*nxxxx* keyword values is primarily their responsibility. However, these teams are encouraged to refer to pre-existing calibration datasets within the caldb and to the requirements of downstream s/w tasks prior to delivery to the HEASARC. Perhaps the best rule of thumb concerning the number of `CBD`*nxxxx* limitations is: the more, the better (up to a limit of 9).

3 OVERVIEW OF CALIBRATION DATA TYPES

In the general case, calibration information can be stored as:

- a data array containing the actual (often n-dimensional) calibration datase;
- a subroutine (containing an algorithm) from which a required calibration dataset can be directly calculated.

Within the HEASARC caldb, both methods of storage for calibration information are allowed, with the mandatory character keyword `CDTP`*xxxx* (referred to in Section 2.1) provided for distinguishing between them.

- `CDTP`*xxxx* = `'DATA'`, if the file contains the actual calibration dataset (see also Section 4)
- `CDTP`*xxxx* = `'TASK'`, if the file is a 'Virtual' Calibration File (Section 5) whereby the calibration dataset is constructed by a standalone s/w task.

It should be stressed however, that in the case `CDTP`*xxxx* = `'TASK'`, neither the source code, or executeable of the standalone task is stored within the virtual calibration file itself. Sets of coefficients *etc* for an algorithm from which a required calibration dataset can be calculated and other parameters can also be stored within a virtual calibration file (see CAL/GEN/92-013, George, Zellar & White 1992).

As a result storage of coefficients within a `CDTP`*xxxx* = `'DATA'` calibration file is **not** allowed.

The pros and cons of storing the necessary calibration information in the form of real data and/or virtual files are provided below on a dataset-by-dataset basis

4 USE OF MULTIDIMENSIONAL DATASETS

The calibration data stored in many calibration files is in the form of n-dimensional arrays (*eg* consider a calibration dataset D which has been measured and is to be stored as a function of $n = 4$ variables W,X,Y & Z at i,j,k & l values respectively; the array D thus contains a total of $i \times j \times k \times l$ elements). Almost without exception, it is necessary that the values of the n parameters (thus defining the $i \times j \times k \times l$ grid-points of the calibration dataset) also be stored — for clarity and to allow interpolation. In order to make such datasets self-contained, to facilitate access and to remove the possible of ambiguity, arrays containing the values of each of the parameters on which that calibration dataset depend are therefore contained alongside the n-dimensional array containing the calibration dataset itself. In order to reduce repetition (of the parameter arrays) and hence save disk-space, for n greater than 2, a BINTABLE FITS format is usually adopted using the 'Multidimensional Array' convention described in Appendix A of Cotton & Tody (1991).

Thus, in the most simple case, the above calibration dataset D would be stored as a BINTABLE with 4 (1-d array) columns containing the i values of W, j values of X, k values of Y and l values of Z. The final column would then consist of the 4-dimensional array containing the $i \times j \times k \times l$ values of D. In cases where a parameter at a grid-point cannot be represented by a single value (since it is the result of an integration/measurement over a range of values — photon energy is the classic example), then additional columns are required. In the above example the column containing the i values of parameter A (say) would therefore be replaced by 2 columns containing the i values of A_{low} and A_{high} respectively (or whatever is more appropriate). Finally, in the most general (but rare) case, some or all the columns containing information on the parameter (grid point) values may themselves be n-dimensional arrays.

4.1 Ordering of Multidimensional Datasets

The ordering of the data values within all n-dimensional calibration datasets (with n greater than one) is of course crucial. Following the "Multidimension Array" convention of Cotton & Tody (1991), all

> [...] *columns within a* BINTABLE *will have an associated character keyword* `TDIM`*nnn = '(i,j,k,...)' where i,j,k,... are the dimensions of the array,* [and *nnn* is the number (*nnn = 1, 2, 3, ..., 10, 11, ... 100, 101... etc*) of the column in which the array is stored]. *The data is ordered such that the array index of the first dimension given (i) is the most rapidly varying and that of the last dimension given in the least rapidly varying. The size implied by the* `TDIM`*nnn keyword will equal the element count specified in the* `TFORM`*nnn keyword. The adherence to this convention will be indicated by the presence of a* `TDIM`*nnn keyword in the form described above.*

Thus, if the example calibration dataset D referred to in the previous section had i=1024, j=20, k=30 & l=2 was stored in the 5$^{\text{th}}$ column with parameter i changing fastest, k the next fastest, j the next fastest, and l the slowest, then `TDIM5` = `'(1024,30,20,2)'`.

Within the HEASARC caldb, the `TDIM`*nnn* keyword is mandatory for all n-dimensional datasets stored within a column of a single row BINTABLE.

The ordering of the parameters within the multidimensional calibration datasets described below has been chosen to facilitate access, and specifically to minimize access time. The ordering has therefore been determined on a calibration datatype-by-datatype basis, and is described in more detail for each individual datatype below. It is strongly recommended that a summary of what the ordering means physically for all n-dimensional arrays are also given via a liberal use of `COMMENT` keywords.

4.2 The 'axis labels' for multidimensional columns

The type of physical quantity represented by the axis of all n-dimensional datasets (*ie* the 'axis labels') should be explicitly stored using the *i*`CTYP`*nnn* keywords of OGIP/94-006 (Pence *et al.* 1994), where i is the dimension (from fastest to slowest varying). The definitions and allowed values for the *i*`CTYP`*nnn* keywords are given in Table 2.

Table 2: Allowed *i*CTYP*nnn* keyword values

i*CTYP*nnn **String**	**Description** Space along which axis is in ...
`ENERGY`	Energy
`CHANNEL`	(detector) Channel
`COORD-`X	(spatial) Coordinate (dimension X, where $X =$ `1`, `2` *etc*

where:
nnn is the column number of the n-dimensional column, $nnn =$ `1`, `2`, `3`, ... *etc*
i is the dimension of the axis

Thus in the above example, if the 4-dimensional dataset in column 5 represented a $1024 \times 20 \times 30 \times 2$ array of ENERGY vs THETA vs PHI vs TIME (respectively), the corresponding family of keywords would be:
`1CTYP5` = `'ENERGY'`
`2CTYP5` = `'COORD-1'`
`3CTYP5` = `'COORD-2'`
`4CTYP5` = `'TIME'`

4.3 Cross-referencing 'axis grid' with columns

The axes of an n-dimensional dataset should be cross-referenced to the names of the columns (in the **same extension**) containing the grid-point values using the `CREF`*nnn* keyword. This keyword has a relatively straightforward syntax, but which is best demonstrated with an example. So, continuing the above example, if the grid-points of the dataset in column 5
– in Energy-space are stored in columns named 'ENERG_LO' and 'ENERG_HI' (for the lower and upper energies of each bin);
– the grid-points along the 1st spatial coordinate are stored in a column named 'THETA' (where the grid are 'point-measurements', meaning the lower & upper angles for each bin are indentical);
– the grid-points along the 2nd spatial coordinate are stored in a column named 'PHI' (again 'point-measurements);
– and the grid-points along the temporal -dimension are stored in columns named 'START' & 'STOP', then the correct syntax would be:
`CREF5 = '(ENERG_LO:ENERG_HI,THETA,PHI,START:STOP)'` It should be noted that the dimensions **must** be listed in the same order as for the n-dimensional dataset itself.

4.4 Number of elements in 1-d arrays

In BINTABLE extensions consisting of a single row, the number of elements in a 1-d array stored in a given column *nnn* is of course specified using the (FITS) mandatory keyword `TFORM`*nnn*.

4.5 Specifying Columns as Keywords

There is a general convention used throughout the HEASARC caldb

whereby if a column in a FITS data table contains the same (single) entry in every row, then that column can be deleted from the table and specified instead as a keyword. Downstream s/w should thus look first for a keyword of the required name; then, if the keyword is not found, search for a similarily named column.

4.6 Unnecessary Columns & Keywords

The formats summarized below cover the most general (multi-instrument) cases. Thus in the application of a given format to a specific instrument dataset, certain columns/keywords of the FITS data table may be inappropriate or thought unnecessary. However it should be remembered that in most cases the downstream s/w will have been designed with the general case in mind. Hence problems/ambiguities may result if such values are not specified. In most cases this simply involves the straightforward specification of a keyword (see also Section 4.5).

Obviously those columns and keywords listed below as optional do not have to be specified if not appropriate or required.

5 THE USE OF VIRTUAL CALIBRATION FILES

'Virtual' calibration files have been designed to address the case where a given calibration dataset can more easily and economically be expressed as an analytical function. Rather than storing the actual calibration data as a n-dimensional array, a virtual calibration file contains the name & location of a standalone task as FITS keywords and, within the 'data array', any additional information (*eg* parameters, coefficients *etc*) required to execute that task. When the calibration data is required by downstream s/w, the task is spawned and executed from the main program, the required calibration dataset generated and passed back to the main program.

Virtual calibration files will reside alongside 'real' (`CDTP`*xxxx* = `'DATA'`) calibration files within the HEASARC caldb, and are also represented as entries in the appropriate CIFs (Section 2).

Virtual calibration have thus been designed to fulfill the following roles:

- to simplify the task of downstream s/w given the potential mix of virtual and 'real' calibration datasets (describing the same type of calibration information),
- to inform users (via CIFs) that a given calibration dataset is produced by a standalone s/w task, rather than existing as data already on disk.
- to keep track of multiple versions of virtual calibration datasets (*ie* employing slightly different algorithms for the same calibration information), and to allow users to choose between then if they so wish.

A detailed description of the contents, format and operation of Virtual Calibration Files, and the associated standalone tasks is given (CAL/GEN/92-013 (George, Zellar & White 1992).

5.1 Mandatory Keywords for Virtual Calibration Files

In addition to the standard FITS keywords, and those required for CIF purposes (Section 2.1), the following keywords are also required for Virtual Calibration Files:

- `CTASKLOC` - the disk location of the executable of the standalone task. In HEASARC-supplied datafiles, this will be an environment variable giving the path to the executeable (see CAL/GEN/92-013; George, Zellar & White 1992).
- `CTASKNAM` - the name of the standalone task to be executed (including any extension such as '`.exe`').
- `CTASKLAN` - the computer language in which the standalone task is written

6 PHYSICAL UNITS OF DATASETS

In order to facilitate downstream s/w use of the calibration data, it is clearly crucial that a common set of character strings is used to specify the physical units of datsets throughout all calibration (and other) datafiles and software. A list those current approved for use within the HEASARC caldb is provided in OGIP/93-001 (George & Angelini 1993).

Authors of calibration datasets and s/w are reminded that a character string specifying the physical units of a parameter is required in two places within calibration dataset:

- For the value of the (FITS) mandatory `TUNIT`*nnn* keyword for column *nnn* in a BINTABLE extension,
- For use with the `CBD`*nxxxx* keyword recommended/required for CIF purposes (see Section 2.2).

7 PHOTON ENERGY GRIDS

The following rules and conventions apply to all arrays containing (photon) energy within the HEASARC caldb:

- It is strongly recommended that wherever possible the same energy bins be used for all energy-dependent calibration datasets for a given instrument *ie* that a standard energy grid be used.
- Seperate arrays containing the lower (E_{low}) and upper bound (E_{high}) should be specified (rather than (say) the mean energy and half-width).
- Values of both E_{low} & E_{high} are generally given for each energy bin for clarity, and to ease access & use. The order should be sequential, starting from the lowest value. In no case should there be any overlap between consecutive energy bins, *ie* $E_{low}(i) \geq E_{high}(i-1)$, (although in most cases, $E_{low}(i)$ will equal $E_{high}(i-1)$).
- The Energy grid should be of sufficient resolution for use with all downstream s/w. In particular, since the size ($E_{high}(i) - E_{low}(i)$) of the individual bins need not be uniform, very high resolution (small bin sizes) close to sharp features such as edges *etc* should be employed.
- If necessary **downstream s/w will simply perform a linear interpolation in any gap between adjacent energy bins**, *ie* in cases where $E_{low}(i) > E_{high}(i-1)$.

8 THE SPECIFICATION OF TIMES

A number of calibration datasets are an explicit function of time (excluding datasets in which errors are discovered requiring updates). An example is a Bad Pixel map (CAL/GEN/92-026; George & Zellar 1993a) dataset which lists each pixel which is considered to have 'gone bad', and the time at which it was deemed to do so.

All such calibration files conform to the HEASARC standards for the specification of times outlined in OGIP/93-003 (Angelini *et al.* 1993). The following keywords are therefore mandatory for all calibration datasets including times (*ie* a column containing the time):

- `TIMESYS` - the time frame used to specify all times within the dataset.
  The following values are allowed:
  - `TIMESYS` = `'JD'`, for Julian Day
  - `TIMESYS` = `'MJD'`, for modified Julian Day (MJD = JD - 2400000.5)
  - `TIMESYS` = `'TJD'`, for modified Julian Day
  - `TIMESYS` = `'1990.00'` (*etc*) for an arbitrary reference date (specification in decimal years only is allowed)
- `TIMEZERO` - the 'zero-offset' of the times stored in the data table.

- `TIMEUNIT` - the unit of time used for the value of the `TIMEZERO` keyword
- `CLOCKCOR` - whether the time given by the value of `TIMEZERO` (in the system defined by `TIMESYS`) has been corrected for any drift in the s/c clock relative to UT. This keyword is useually overkill in the case of calibration datasets, but is included as it is mandatory in OGIP/93-003 (Angelini *et al.* 1993).

and the following keyword recommended:

- `MJDREF` - the reference time adopted for each mission in MJD. Different missions often have reference times defined using different (but convenient) unit systems. The reference time is the only time information which is not affected by any mission-specific corrections.

9 ERRORS ON DATASETS

Errors have traditionally not been supplied as part of calibration datasets. However, in order to make the formats as general as possible, all BCF & CPF datasets within the HEASARC caldb have the **optional** facility to include error information. H/w teams *etc* supplying calibration data to the HEASARC caldbare urged to quantify and supply the errors associated with each dataset, if for no other reason, to allow users to assess the accuracy of the calibration.

There are two types of possible errors:

- statistical/measurement errors
- fractional systematic errors (relative to the corresponding data value)

In most cases errors are quantified within HEASARC calibration files using the 68% confidence levels (*ie* standard errors for measurements distributed according to Gaussian statistics). Thus the total error on any element within the calibration dataset is given by the sum of the corresponding statistical and (calculated actual) systematic errors added in quadrature. Exceptions to this are noted in the corresponding definition of the file format and/or within the FITS Header of the dataset concerned, and in the corresponding calibration documentation for that dataset.

The ordering of the data within all such error arrays must be the same as that for the calibration dataset to which it refers. (For single row BINTABLEs the ordering is specified by the mandatory `TDIM`*nnn* keyword as described in Section 4.1.) It is strongly recommended that the units used to specify the statistical errors are the same as those used for the calibration dataset to which it refers (**required** if the column is removed and specified as a keyword — see Section 4.5).

In order to accomodate both symmetic and assymetric errors, in the general case the size of both the negative and positive errorbars of both types are given in seperate data arrays. Unfortunately, due to the 8-character limit on keyword name provided by FITS (important for reasons described in

Section 4.5), no totally uniform or consistent naming convention for the necessary column names for all calibration datasets is possible. In the rest of this section we use `STAT_MIN` & `STAT_MAX` to denote column names of the negative and positive statistical errors (respectively), and `SYS_MIN` & `SYS_MAX` to denote the same for the fractional systematic errors. The corresponding FITS column names for the individual data formats are described in the relevant sections below.

Since it is not always necessary/possible to specify all/any of the errors described above, downstream s/w should use the following logic if the errors of a dataset are required:

1. search for columns with names `STAT_MIN` and `STAT_MAX`.
2. if both exist, then both negative and positive statistical errors are available for this dataset.
3. otherwise, if only one of the above columns exists, then search for the other as a header keyword
    - if the other exists as a keyword, then its value is constant for all elements of the calibration dataset (conforming to the convention outlined in Section 4.5). Thus both negative and positive statistical errors are available for this dataset.
    - otherwise, if only a `STAT_MIN` column exists (and `STAT_MAX` is not supplied as a keyword), then the negative and positive statistical errors can be assumed to be symmetric, and their size are stored in the `STAT_MIN` column.
4. repeat steps 1 – 3 for the systematic errors `SYS_MIN` and `SYS_MAX` (with the same rules regarding keyword searches/presence)
    - if the fractional systematic errors are available for a given dataset, then they should be multipled by the corresponding data element, and (if required) added in quadrature with any corresponding statistical error to give the total error on each data value.
5. if none of the above are supplied as either columns or keywords then no error information exists for this dataset.

If should be emphasized that it is rare that both types of errors exist for a given calibration dataset, and even more rare that this information is required by downstream s/w.

10 HDUCLASn & HDUVERSn KEYWORDS

The HEASARC FITS Working Group (HFWG) has long been considering a proposal to provide a hierarchical classification scheme for the various types of FITS extensions used within the HEASARC. It is proposed that this be achieved using:

- keywords of the form `HDUCLAS`n (where n is an index, with n=1 being the highest level of the hierarchy, and n=9 the lowest) to store character strings representing the scientific contents of the extension, and

- corresponding keywords of the form `HDUVERS`*n* to store the version number of the actual format used (but still as character strings)

A number of HEASARC-produced files have already adopted this system, and if raised to a full recommendation, it is anticipated all HEASARC-produced files will be expected to quickly follow. Calibration files prove no exception to this.

At the current time, the allowed `HDUCLAS`*n* & `HDUVERS`*n* values are listed as part of the general description of the individual file formats.

11 COORDINATE FRAMES

There are usually a large number of coordinate systems used in a given instrument and s/c (see Table 3). Many calibration datasets are a function of position, usually specified in either the mirror assembly/collimator coordinate frame, or one of the (often cartesian) coordinate frames of the detector. Careful definition and specification of the coordinate frame in use is obviously required to ensure a correct application of the calibration dataset to the science data. A straightforward example is provided by the numerous detector coordinate frames which can result following the removal of the various distortions due to the operation of the detector.

In order to facilitate the automatic s/w checks that consistent coordinate frames are indeed being used for both the science and calibration data (and to clarify to users exactly what frame is in use), the FITS (character) keyword `CSYSNAME` is mandatory for all calibration datasets stored in FITS table format listing spatial coordinates. The value of this keyword can thus be used by downstream s/w to directly determine whether any coordinate transformations are required to be applied to the stored calibration datasets prior to comparision with the science data. It should be noted that in all cases the definition of the respective reference frames, and documentation of the coordinate transformations between frames is the responsibility of the h/w groups.

11.1 Allowed CSYSNAME values

The following values of the `CSYSNAME` are currently in use associated with calibration datasets:

- `CSYSNAME = 'RAW_DET'`
  Raw (usually pseudo-cartesian) detector coordinates (pixels), as output by the on-board electronics and/or any standard pre-processing.
  In the file formats described throughout the rest of this memo, the column names/keywords `RAWX` & `RAWY` are used to denote the values/arrays containing the stored positional grid-points within a calibration dataset which uses this coordinate frame (but see Sections 11.3 & 11.2). The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
  The definition of the coordinate frame (reference point and definition of the axes) is detector-specific and should be clearly documented for each detector.

Such pixel coordinates are dimensionless, but in cases where the raw detector pixels have been binned up (blocked), the blocking factor relating the stored values to the intrinsic detector pixel size is specified by the mandatory `BLCX`*nnn* & `BLCY`*nnn* keywords (see Section 11.2).

- `CSYSNAME = 'LIN_DET'`
  Detector coordinates corrected for all spatial distortions (due to detector electronics, geometry *etc*) and transformed onto a cartesian grid perpendicular to the optical axis of the instrument. These coordinates ought easily be able to be transformed into position on the sky relative to the optical axis and one other orthonal vector. Thus, for imaging instruments, this is the coordinate frame most commonly used by observers to display scientific images.
  In the file formats described throughout the rest of this memo, the column names/keywords `DETX` & `DETY` are used to denote the values/arrays containing the stored positional grid-points within a calibration dataset which uses this coordinate frame (but see Sections 11.3 & 11.2). The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
  The definition of the coordinate frame (reference point and definition of the axes) is detector-specific and should be clearly documented for each detector.
  Such pixel coordinates are dimensionless, but in cases where the raw detector pixels have been binned up (blocked), the blocking factor relating the stored values to the intrinsic detector pixel size is specified by the mandatory `BLCX`*nnn* & `BLCY`*nnn* keywords (see Section 11.2).

- `CSYSNAME = 'PHY_DET'`
  Physical detector coordinates (in units of length) as measured on the surface of the detector/window/filter.
  In the file formats described throughout the rest of this memo, the column names/keywords `PHYX` & `PHYY` are used to denote the values/arrays containing the stored positional grid-points within a calibration dataset which uses this coordinate frame. The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
  The definition of the coordinate frame (reference point and definition of the axes) is detector-specific and should be clearly documented for each detector.
  The physical units of the stored measurements should be specified by the value of the relevant `TUNIT`*nnn* keyword as normal. The recommended units are mm.

- `CSYSNAME = 'XMA_POL'`
  Polar coordinates relative to the optical axis of the X-ray mirror assembly/collimator and one orthogonal vector.
  In the file formats described throughout the rest of this memo, the column names/keywords `THETA` & `PHI` are used to denote the values/arrays containing the stored off-axis & azimuthal angles (respectively) grid-points within a calibration dataset which uses this coordinate frame (but see Section 11.3). The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
  The reference point for this coordinate frame is the projection of the optical axis onto the surface of the detector. The definition of the orthogonal reference vector (*ie* where $\phi = 0.0$) is detector-specific and should be clearly documented for each detector.
  The physical units of the stored maeasurements should be specified by the value of the relevant `TUNIT`*nnn* keyword as normal. The recommended units are arcmin.

- `CSYSNAME = 'XMA_CART'`
  Orthogonal angles measured from the optical axis of the X-ray mirror assembly/collimator along

two mutually orthogonal vectors.
In the file formats described throughout the rest of this memo, the column names/keywords `ALPHA` & `BETA` are used to denote the values/arrays containing the stored angle grid-points within a calibration dataset which uses this coordinate frame — `ALPHA` being measured along the same (focal plane) reference vector used in the `XMA_POL` system used above, and `BETA` being measured along the orthogonal (PHI = 90.0 degree) vector. The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
The reference point for this coordinate frame is the projection of the optical axis onto the surface of the detector. The definition of the reference vectors is detector-specific and should be clearly documented for each detector.
The physical units of the stored maeasurements should be specified by the value of the relevant `TUNIT`*nnn* keyword as normal. The recommended units are arcmin.

- `CSYSNAME = 'XMA_DCOS'`
  Direction cosines measured from the optical axis of the X-ray mirror assembly/collimator along two mutually orthogonal vectors.
  In the file formats described throughout the rest of this memo, the column names/keywords `COS_ALPHA` & `COS_BETA` are used to denote the values/arrays containing the stored angle grid-points within a calibration dataset which uses this coordinate frame — `COS_ALPHA` being measured along the same (focal plane) reference vector used in the `XMA_POL` system used above, and `COS_BETA` being measured along the orthogonal (PHI = 90.0 degree) vector. The number of such grid-points is specified using the `TDIM`*nnn* keyword as outlined in Section 4.4.
  The reference point for this coordinate frame is the projection of the optical axis onto the surface of the detector. The definition of the reference vectors is detector-specific and should be clearly documented for each detector.
  The stored measurements are unitless.

- `CSYSNAME = 'DET_MSK'`
  ... *section incomplete* special coordinates used to facilitate the specification of any spatially non-uniform filter or window support structure (*eg* a wire mesh)

These are briefly summarized in Table 3, and will be added to as necessary.

Clearly all coordinate systems are not appropriate to all instruments or calibration datasets. The file formats for storing the (calibration) information necessary to transform between the various coordinate frames is discussed in OGIP/92-016 (George & Yusaf 1992b), and the recommended/usual coordinate frames for each of the other calibration datasets are listed in the relevant section below.

11.2 Pixel Definitions

Within the HEASARC caldb, the standard FITS convention has been adhered to whereby the locations of all pixels are defined by the centre of that pixel measured from the lower left corner of the image.

Thus in cartesian coordinates x, y, the pixel (i, j) is located in the region in which $i-1/2 \leq x \leq i+1/2$ and $j - 1/2 \leq y \leq j + 1/2$. The standard column names/keywords `DETX` & `DETY` are used to denote

Table 3: Notation for various spatial coordinates used within the HEASARC caldb

CSYSNAME value	**Coordinate Frame**	**Description**
spatial coordinate frames used for calibration datasets		
`RAW_DET`	Raw detector	as determined by on-board detector electronics (uncorrected in any way)
`LIN_DET`	Linearized detector	all positional distortions due to detector removed
`PHY_DET`	Physical Detector *etc*	in units of length on the detector/window/filter surface
`XMA_POL`	Optical Axis	as defined by optical axis of mirror/collimator and orthogonal reference vector
`XMA_CART`	Optical Axis	as above, but in using cartesian system
`XMA_DCOS`	Optical Axis	as above, but in using direction cosines

the X & Y pixel coordinates in all calibration datasets requiring this information within this memo (but see Section 11.3).

The size of (each side of) a pixel relative to the 'raw' (intrinsic) pixel size of the instrument (*ie* the 'blocking factor') is specified via the keywords `BLCX`*nnn* & `BLCY`*nnn* (where *nnn* is the column number of the corresponding coordinate). The absence of these keywords should be taken to imply that the pixels are not blocked. Thus in the above example, if `BLCX`$nnn = M$ & `BLCY`$nnn = N$, then the top-left corner of the blocked pixel (i,j) is at $x_{raw} = (i - 1/2) \times M$, $y_{raw} = (j + 1/2) \times N$ in raw pixel coordinates.

The location of the origin, along with the definition & orientation of the coordinate system and the specification of the 'raw' pixel size are the responsiblity of the h/w team.

11.3 Naming Convention for Columns Describing Spatial Coordinates

In the above definitions of the various coordinate frames (Section 11.1) standard column names/keywords were defined for use within the HEASARC caldb, datasets (*eg* `DETX`, `DETY`, `THETA`, `PHI` *etc*). In conjuction with the value of the relevant CSYS*nnn* keyword, these define explicitly the spatial coordinate frame used in all calibration files.

However (unfortunately), a number of the coordinate frames for some instruments, as defined by the h/w teams, do not use the standard axis notation assumed here. For example, an instrument may employ a left-handed detector coordinate system in which the X- and Z-axes define the detector plane (rather than the right-handed system using the X- and Y-axes assumed above). In order to avoid ambiguities and confusion, all datasets in the HEASARC caldb, will use the notation adopted by the project (*ie* the `DETX`, `DETY` *etc* column names will be replaced by the corresponding alternatives: `DETX` & `DETZ` in this example).

This convention, however, gives rise to an obvious problem for multi-mission analysis & calibration s/w: what are the names of the columns containing the spatial coordinates ? In order to minimize the need for mission-specific routines in downstream s/w tasks, the mandatory keywords `CSYSNAME` & `CREF`*nnn* are used. The value of the `CSYSNAME` keyword is a character string which lists (in the order X, Y or θ, ϕ) the non-standard column names/keywords containing the spatial arrays separated by a comma (*eg* in the above example `CSYSNAME = 'DETX,DETZ'`). Similarly the rules governing the cross-referencing of the axes of an n-dimensional dataset in column *nnn* to the names of the columns storing the grid-points described in Section 4.3 lead to the non-standard column names also appearing in the values of the `CREF`*nnn* keyword. The names of the `BLCX`*nnn* & `BLCY`*nnn* keywords giving the blocking factor of an image should **NOT** be changed in these cases.

The lack of the `CSYSNAME` keyword indicates that the standard column names/keywords are used.

ACKNOWLEDGEMENTS

We thank the numerous people who have contributed ideas, suggestions and constructive critism of both this document (and more importantly) the standard formats described within. In particular we are indebted to the members of the *ASCA* GOF at NASA/GSFC (Charles Day, Eric Gotthelf, Koji Mukai & Ken Ebisawa), to Mike Corcoran, and as always to Bill Pence.

REFERENCES

Angelini, L., Pence, W. & Tennant, A.F., 1993. *Legacy*, **3**, 32. (OGIP/93-003)
Cotton, W.D. & Tody, D., 1991. In preparation.
George, I.M. & Angelini, L., 1993. *Legacy*, **4**, in press. (OGIP/93-001)
George, I.M. & Yusaf, R., 1992a. *OGIP Calibration Memo* CAL/GEN/92-020
George, I.M. & Yusaf, R., 1992b. *OGIP Memo* OGIP/92-016 (in preparation)
George, I.M. & Zellar, R.S, 1993a. *OGIP Calibration Memo* CAL/GEN/92-026
George, I.M., Pence, W. & Zellar, R.S., 1992. *OGIP Calibration Memo* CAL/GEN/92-008
George, I.M., Zellar, R.S, & Pence, W., 1992. *OGIP Calibration Memo* CAL/GEN/92-011
George, I.M., Zellar, R.S, & White, N, 1992. *OGIP Calibration Memo* CAL/GEN/92-013
Greisen, E.W. & Harten, R.H., 1981. *Astron. Astrophys. Suppl.*, **44**, 371.
Grosbol, P., Harten, R.H., Greisen, E.W. & Wells, D.C., 1988. *Astron. Astrophys. Suppl.*, **73**, 359.
Harten, R.H., Grosbol, P., Greisen, E.W. & Wells, D.C., 1988. *Astron. Astrophys. Suppl.*, **73**, 365.
Pence, W., Rots, A., Angelini, L. & McDowell, J., 1994. *Naming Conventions for FITS Coordinate System Keywords* (OGIP/94-006)
Schlesinger, B., 1990. *A User's Guide for the Flexible Image Transport System.* (Version 2.0).
Wells, D.C., Greisen, E.W. & Harten, R.H., 1981. *Astron. Astrophys. Suppl.*, **44**, 363.

All the OGIP memos are available on-line from the anon ftp account on `legacy.gsfc.nasa.gov`