Calibration Memo CAL/GEN/92-008

# Calibration Index Files

Ian M George

Bill Pence

& Ron S Zellar
Codes 664 & 668,
NASA/GSFC,
Greenbelt,
MD 20771

Version: 1995 Mar 01

## SUMMARY

The role and detailed format of Calibration Index Files (CIFs) within the Calibration Database (caldb) are discussed. The associated s/w tasks `crcif` & `udcif` are also briefly described, as are a number of other tasks and FORTRAN calibration subroutine library (`callib`) tasks involved in the access of CIFs. References are provided to documentation for the installation and maintenance of Calibration Database (caldb) s/w & data at remote sites.
Intended audience: primarily authors of downstream s/w, general users of OGIP calibration data at remote sites.

## LOG OF SIGNIFICANT CHANGES

| **Release Date** | **Sections Changed** | **Brief Notes** |
|---|---|---|
| 1993 Feb 22 | | Original Version |
| 1995 Mar 01 | All | Made compatible with `LaTeX2HTML` software |
| 2004 Apr 01 | All | Made compatible with tth |

# 1 INTRODUCTION

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

Due to various logistic reasons, and as an aide to clarity, it is proposed that the calibration data is stored in a relatively large number of small files within the caldb. Broadly speaking, each file will be dedicated to a single aspect of calibration, and consist of a single FITS extension containing a single calibration dataset. In some cases, however, a few closely related datasets may be combined in a single FITS extension in order to save disk space, and for clarity. In the event of a calibration update, the new calibration information will be written into a new file, which (wherever possible) employs the same format corresponding old calibration file. The old file, however, will of course be retained within the caldb and accessible to users and downstream s/w.

The advantages of such a scheme are:

- the individual files are relatively small. This reduces the time taken for the FITS reader to find the relevant dataset within the file, and makes the files more manageable within the database.
- in the case of a update to a single calibration dataset, clearly only the affected file need to transported to remote users, reducing the load on the networks.
- certain Basic Calibration Files (BCFs; see George 1992) may in fact be applicable to more than one instrument reducing repetition. For example, the BCFs associated with the *ROSAT* XRT mirror assembly will be applicable to both the PSPC and HRI instruments.
- the retention of the old calibrations allows users the flexibility in comparison of the effect the various calibrations makes to their scientific results.
- in the medium term all calibration files within the OGIP Calibration Database (caldb) will be stored on write-only optical disk, thus the deletion of previous versions of a given calibration dataset will in any case be impossible.

whilst the main disadvantage is:

- the large number of calibration files.

Strict configuration control and database management is therefore clearly required. The FITS header of each calibration file will contain all neccessary details concerning the contents, orgin, time-tagging for use, and (to a limited extent) information on the use of the data within. Such information will also be given in greater detail in relevant documentation (*eg.* the 'OGIP Calibration Guide' for each instrument). However, standard information concerning the name, location, format and brief summary of contents of each calibration dataset within each file within the OGIP caldb will also be stored in "Calibration Index Files". It is intended that these files play a crucial role in this file-housekeeping both within the OGIP and at a remote user's home institution. They therefore form the subject of this memo.

In Section 2 the design requirements of CIFs are described, along with a brief discussion of their role within the OGIP Calibration Database (caldb) and their interaction with downstream s/w. A detailed description of the format of a CIF file is given in Section 3. Then, in Section 4 we briefly review the requirements (mandatory keywords) imposed on all calibration files to be included within the OGIP caldb. In Section 5 a number of associated s/w tasks and routines are listed. Finally in Section 6, references are provided to other documentation which detail the creation and maintenance of CIFs, both within the OGIP caldb, and at remote sites.

# 2 THE PURPOSE OF CALIBRATION INDEX FILES

A primary goal of the HEASARC is to make all s/w as mission/detector independent as possible. To this end, i/p filenames *etc.* are not hardwired into the code. In the case of calibration data, this policy obviously gives users & s/w greater flexiblity in choosing which calibration data they wish to use. Of course in most cases, users will desire to use the standard (best) calibration data, and this will be the default case. It will be the responsibility of the user to ensure any non-default calibration data they use makes sense.

Calibration Index Files (CIF) have been designed to fulfill the following functions:

1. To enable down-stream s/w to locate required calibration information.
2. To enable the HEASARC to keep track of the numerous calibration files within the archive.
3. To provide users with a concise summary of the available calibration information. (A copy of the information within the CIF will also be provided on-line at the HEASARC, accessible through the XOBSERVER account, enabling users to see (BROWSE) what's available.)

It is anticipated that the majority of GOF/HEASARC will wish to perform the bulk of their analysis on their own machines. These users will thus need to copy some/all the OGIP Calibration Database (caldb) to their own machine, along with any required HEASARC software. It is therefore intended that CIFs also perform the above functions for users at their home institution.

## 2.1 Design Requirements & Constraints

The CIFs were designed with the following requirements in mind:

1. The format of the file must be machine independent, and user-friendly (*ie.* FITS).
2. The format must contain all the information necessary for s/w to locate the required file and calibration dataset stored within.
3. The CIFs should be able to be stored anywhere on disk (so as to reduce the restrictions imposed on remote users wishing their own copy of the OGIP Calibration Database (caldb) and associated software).

In order to simplfy s/w and user access, the following constraints have been imposed:

1. A single CIF must contain all the information concerning the calibration files associated with a given instrument (though several instruments may share the same CIF if it is so desired).
   (This is equivalent to saying that all the calibration files associated with a given instrument must reside under a single directory tree, though several instruments can share the same tree).

## 2.2 Overview of Access to & Role of CIFs

### 2.2.1 System Concerns

In order to reduce disk-space and maintenance overheads, it is strongly recommended that remote users on clustered machines share a single local caldb, and that a single person be responsible for it's maintenance.

1. It is strongly recommended (especially on clustered machines) that a system-wide symbolic link (under unix/ultrix) or logical (under VMS) be defined pointing to the physical location of the local caldb sub-directories. Within the OGIP caldb, the 'logical' `caldb` is used for this purpose. Remote users are free to choose their own 'logical', but may wish to adopt the same convention as the HEASARC thus
   - Under unix/ultrix it is necessary to remote mount the filesystem on which the local caldb resides on each machine, then make a symbolic link from `/caldb` (and/or `/CALDB`) to point to this filesystem on each machine.
     *eg.* `ln -s /disk4/mary/mungo/midge/calibration /caldb`

   (This is therefore an identical setup to that recommended to define the `XANADU` 'logical' at remote nodes)

Table 1: Example of contents of cinitu.vms and cinitu.csh files required for CIFs

*under VMS* (`cinitu.vms`)

```
        ........
    ASS caldb:[data.astrod.gis]caldb.indx asdgiscif:
    ASS caldb:[data.astrod.sis]caldb.indx asdsiscif:
    ASS caldb:[data.astrod.xrt]caldb.indx asdxrtcif:
    ASS caldb:[data.rosat.pspc]caldb.indx rospspccif:
    ASS caldb:[data.rosat.hri]caldb.indx roshricif:
                              etc.
        ........
$
```

*under the unix/ultrix C shell* (`cinitu.csh`)

```
        ........
     setenv asdgiscif  /caldb/data/astrod/gis/caldb.indx
     setenv asdsiscif  /caldb/data/astrod/sis/caldb.indx
     setenv asdxrtcif  /caldb/data/astrod/xrt/caldb.indx
     setenv rospspccif /caldb/data/rosat/pspc/caldb.indx
     setenv roshricif  /caldb/data/rosat/hri/caldb.indx
                              etc.
        ........
```

2. The disk location and name of the various CIFs is stored using environment variables defined in a site-specific (calibration) initialization file (`cinitu.csh` under the unix C shell). Examples of the contents of this file relevant to CIFs are given in Table 1 (in which all CIFs have the same name, `caldb.indx`, for calrity). This file must be executed prior to using any calibration s/w (and will be sourced by the appropriate `go_caldb` file). It is the responsibility of the remote user with a local caldb to ensure that the appropriate pointers within this file are defined correctly. This is discussed in more detail in CAL/GEN/92-015 (George & Zellar 1993a).

### 2.2.2 User Access

The CIFs will be in FITS format and hence readily accessible to users (via FTOOLS or a similar general FITS reader) wishing to inspect all the calibration datasets currently available.

Obviously, once the location and name of the desired calibration dataset has been obtained, users are able to use other FTOOLS *etc.* to inspect, extract & manipulate the data file themselves.

#### 2.2.3 Software Access to Calibration Datasets and the Role of CIFs

Software access to calibration datasets via CIFs is as follows:

1. A CALTOOLS task `quzcif` (or `callib` subroutine library routine `qzcif`, see Section 5.3.1), is called from the main s/w task. This finds and opens the required CIF, searchs it for the requested (passed) entry, and if found, returns the name & disk location of the file containing the required calibration information. A more advanced mode of operation is also possible in which the user can be prompted for which version of a given calibration dataset they wish to use – see Section 5.3.1.
2. The main s/w task then finds and opens this calibration file, and hence is able to access the extension containing the required calibration dataset.

In most cases, higher level tasks within the OGIP calibration FORTRAN subroutine library (`callib`) are available to perform the above 2 steps (see Section 5.3.2).

# 3 CALIBRATION INDEX FILE FORMAT

The Calibration Index Files consist of a FITS file with a null primary array and a single BINTABLE extension. Each row within the extension refers to a single calibration dataset within a single extension in a single calibration file. The values inserted into the various columns of the BINTABLE originate either from the calibration file or are supplied by the installation s/w (`crcif`, `udcif`) as noted below. As stated above, when taken together each row of the CIF contains all the information for downstream s/w to identify, locate a required calibration dataset.

## 3.1 The INDEX Extension

### 3.1.1 Extension Header

Besides the standard FITS keywords, the header contains the following (mandatory) keywords/values:

- `EXTNAME = 'CIF     '`, the name of the extension
- `CIFVERSN` - the OGIP version number of the FITS format in use (in this case `CIFVERSN = '1992a   '`)

#### 3.1.2 Data (Table) Format

The BINTABLE of the CIF consists of the following columns:

1. *Telescope*, a 10-byte CHARACTER string specifying the telescope/mission to which the calibration data applies (see CAL/GEN/92-011, George, Zellar & Pence 1993, for a summary of allowed values, available on-line as pdf and html versions).
   The FITS column name is **TELESCOP**, and the value obtained directly from the `TELESCOP` keyword in the calibration file.

2. *Detector*, a 10-byte CHARACTER string specifying the detector/instrument to which the calibration data applies (see CAL/GEN/92-011, George, Zellar & Pence 1993, for a summary of allowed values, available on-line as pdf and html versions).
   The FITS column name is **INSTRUME**, and the value obtained directly from the `INSTRUME` keyword in the calibration file.

3. *Detnam*, a 20-byte CHARACTER string further specifying the specific detector to which the calibration refers (*ie.* when the value of *Detector* is insufficient). CAL/GEN/92-011 (George, Zellar & Pence 1993, available on-line as pdf and html versions) gives a list of instruments for which this keyword is often required along with a summary of allowed values.
   The FITS column name is **DETNAM**, and the value obtained directly from the `DETNAM` keyword in the calibration file (if present).

4. *Filter*, a 10-byte CHARACTER string specifying the filter in use. This column will contain the string 'NONE' in the case of instruments without a moveable filter or for calibration datasets for which filter information is not relevant (see CAL/GEN/92-011, George, Zellar & Pence 1993, for a summary of allowed values, available on-line as and html versions).
   The FITS column name is **FILTER**, and the value obtained directly from the `FILTER` keyword in the calibration file (if present).

5. *Log_Link*, a 20-byte CHARACTER string specifying the system-wide symbolic link (under unix/ultrix) or system-wide logical (under VMS) defining the physical location (disk) of the calibration sub-directories within which the file resides (*ie.* `CALDB`, or whatever, as defined in Section 2.2.1).
   The FITS column name is **CAL_DEV**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

6. *Directory*, a 70-byte CHARACTER string specifying the sub-directory location of the calibration file beyond that given by *Log_Link*. The string will conform to OGIP standards for such directory paths whereby the sub-directory names are specified by a slash ("/"). This is true whatever operating system is in use on either the disk on which the calibration data files resides, or (if different) the disk where the CIF resides.
   The FITS column name is **CAL_DIR**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

7. *File*, a 40-byte CHARACTER string specifying the name of the calibration data file.
   The FITS column name is **CAL_FILE**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

8. $Cal_{class}$, a 3-byte CHARACTER string specifying the OGIP class of calibration file (*ie.* PCF, BCF, CPF *etc.* — see CAL/GEN/92-011, George, Zellar & Pence 1993, for a summary of all allowed values, available on-line as and html versions).
   The FITS column name is **CAL_CLAS**, and the value obtained directly from the `CCLS`$xxxx$ keyword in the calibration file.
   This presence of this keyword also flags that the calibration dataset is to be included within the CIF.

9. $Cal_{type}$, a 4-byte CHARACTER string specifying whether the calibration data consists of real data, or 'virtual' data (*ie.* a taskname and associated parameter inputs — see CAL/GEN/92-013, George, Zellar & White 1993 for further information).
   The FITS column name is **CAL_DTYP**, and the value obtained directly from the `CDTP`$xxxx$ keyword in the calibration file.

10. $Cal_{code}$, a 20-byte CHARACTER string giving the OGIP codename describing the calibration dataset.
    The FITS column name is **CAL_CNAM**, and the value obtained directly from the `CCNM`$xxxx$ keyword in the calibration file (see CAL/GEN/92-011, George, Zellar & Pence 1993, for a summary of allowed values, available on-line as and html versions).

11. $CAL_{Bound}$, an array of 70-byte CHARACTER strings giving the limitations/boundaries (if any) of the calibration dataset.
    The FITS column name is **CAL_CBD**, and the values obtained directly from the `CBD`$nxxxx$ keyword in the calibration file (see CAL/GEN/92-011, George, Zellar & Pence 1993, for a detailed discussion and summary of allowed values, available on-line as pdf and html versions).

12. $Ext_{no}$, a 2-byte INTEGER giving the extension number of the $EXT_{name}$ extension within *File*.

    - $Ext_{no} = 0$, data in the primary data array
    - $Ext_{no} = 1$, data in the $1^{st}$ extension
    - $Ext_{no} = 2$, data in the $2^{nd}$ extension, *etc.*

    The FITS column name is **CAL_XNO**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

13. $Val_{start}\ D$, a 8-byte CHARACTER string giving the UTC date (in the format dd/mm/yy) when these calibration data should first be used.
    The FITS column name is **CAL_VSD**, and the value obtained directly from the `CVSD`$xxxx$ keyword in the calibration file.

14. $Val_{start}\ T$, a 8-byte CHARACTER string giving the UTC time (in the format hh:mm:ss) on day $Val_{start}\ D$ when these calibration data should first be used.

The FITS column name is **CAL_VST**, and the value obtained directly from the `CVST`$xxxx$ keyword in the calibration file.

15. $RefT$, an 8-byte REAL giving the values of $Val_{start}D$ and $Val_{start}T$ converted to modified Julian date for the convenience of access s/w.
The FITS column name is **REF_TIME**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

16. *Qual*, a 2-byte INTEGER specifying the nominal quality of the data.
    (a) $Qual = 0$ if the calibration data quality is considered "good"
    (b) $Qual \neq 0$ if the calibration data is considered "bad", with the option of the integer value specifying the origin of the flag:
        - $Qual = 1$; data considered "bad" by h/w team
        - $Qual = 2$; data considered "dubious" by h/w team
        - $Qual = 3$; data "bad or dubious" by association
        - $Qual = 4$; *spare*
        - $Qual = 5$; data considered "bad" by GOF/HEASARC/User
        - $Qual = -1$; reason for "bad" flag unknown
        - $Qual = -9$; dataset no longer exists (file removed from caldb)

    The FITS column name is **CAL_QUAL**, and the appropriate value inserted by `crcif`, `udcif` *etc.* either via the optional `cif.qual` file (see CAL/GEN/92-015, George & Zellar 1993a) or via a prompt to the user. This column can also be edited by `EDTCIF` (Section 5.3.1).

17. *OGIP Date*, a 8-byte character string giving the date (in the format dd/mm/yy) on which this file was installed in the OGIP calibration database.
The FITS column name is **CAL_DATE**, and the appropriate value inserted by `crcif`, `udcif` *etc.*

18. *Desc*, a 70-byte CHARACTER string giving a brief description the the data stored within this extension of $File$.
The FITS column name is **CAL_DESC**, and the value obtained directly from the `CDES`$xxxx$ keyword in the calibration file. This column can also be edited by `EDTCIF` (Section 5.3.1).

These are summarized in Table 2.

# 4 CALIBRATION FILE REQUIREMENTS

As should be apparent from Section 3.1.2, the following keywords are mandatory in the header of the FITS extension containing all calibration datasets:

- `TELESCOP` - the name of the satellite/mission.

Table 2: Summary of the format for CIFs (CIFVERSN = 1992a)

**Extension** to `caldb.indx`
**Name:** CIF
**Description:** Calibration Index File.
**Format:** BINTABLE

| | *column* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| *contents* | | | | | | | | |
| **Telescope or Mission Name** | **Instr. Name** | **Detector Name** | **Filter Name** | **Environ. Variable to caldb** | **Directory** | **Name of Calibration File** | **OGIP Class of Cal. Dataset** | **OGIP Caldb Datatype** |
| *Telescope* | *Detector* | *Detnam* | *Filter* | *Log_Link* | *Directory* | *File* | $Cal_{class}$ | $Cal_{type}$ |
| *format of each column* | | | | | | | | |
| 10-byte char. | 10-byte char. | 20-byte char. | 10-byte char. | 20-byte char. | 70-byte char. | 40-byte char. | 3-byte char. | 4-byte char. |
| *column name* | | | | | | | | |
| TELESCOP | INSTRUME | DETNAM | FILTER | CAL_DEV | CAL_DIR | CAL_FILE | CAL_CLAS | CAL_DTYP |

| | *column (cont)* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| *contents* | | | | | | | | |
| | | | **Validity Start** | | | | | |
| **OGIP dataset** | **Dataset Limits** | **Extension Number** | **Date** | **Time** | **Ref Time** | **Quality Flag** | **Install. Date** | **Descript. String** |
| $Cal_{code}$ | $Cal_{bound}$ | $Ext_{no}$ | $Val_{start}D$ | $Val_{start}T$ | *RefT* | *Qual* | *CaldbDate* | *Desc* |
| *format of each column* | | | | | | | | |
| 20-byte char. | 70-byte char. array | 2-byte integer | 8-byte char. | 8-byte char. | 8-byte real | 2-byte integer | 8-byte char. | 70-byte char. |
| *column name* | | | | | | | | |
| CAL_CNAM | CAL_CBD | CAL_XNO | CAL_VSD | CAL_VST | REF_TIME | CAL_QUAL | CAL_DATE | CAL_DESC |

- `INSTRUME` - the name of the instrument.
- `DETNAM` - the name of the specific detector (applicable only when the value of the `INSTRUME` keyword is insufficient to uniquely specify the necessary information)
- `FILTER` - the name of the filter in use (not required for instruments without a moveable filter, or calibration datasets for which the filter information irrelevant).
- `CCLS`*xxxx* - the OGIP-class of this calibration file.
- `CDTP`*xxxx* - the code denoting whether the extension contains real or vitual data.
- `CCNM`*xxxx* - the OGIP codename of the extension to be used within CIF to describe the contents (for downstream s/w). This keyword is not mandatory in the case of PCFs.
- `CBD`*nxxxx* - an array of strings (with $n$ arbitrary integers between 1 & 9) giving the parameter limitations of the dataset (*eg.* energy range, off-axis angles *etc.*) used within the CIF to further describe the contents for downstream s/w (in association with the value of the `CCNM`*xxxx* keyword).
- `CVSD`*xxxx* - the UTC date when this calibration data should first be used.
- `CVST`*xxxx* - the UTC time 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.

where *xxxx* is a number of the form `0001`, `0002`, `0003` *etc.* These keywords are further described, along with their allowed values in CAL/GEN/92-011 (George, Zellar & Pence 1993, available on-line as and html versions).

It is the responsiblity of those supplying the files for inclusion in the CALDB to ensure that all other relevant keywords are present and correct, and that sufficient `COMMENT` and/or `HISTORY` keywords are supplied to fully identify and describe the dataset. It is also the responsibility of suppliers of calibration datasets to provide all necessary documentation concerning the origin, description, use and limitations of each dataset to the manager of the HEASARC Calibration Database (caldb). Send e-mail to caldbhelp@olegacy.gsfc.nasa.gov for more information.

# 5 ASSOCIATED SOFTWARE TASKS & ROUTINES

All s/w which creates or accesses CIFs will:

- Be freely available to users (and thus the code should be straightforward and very well commented).
- Be user-friendly.

- Be supported on all 'standard' platforms and operating systems.
- Be able to run both standalone and under the `IRAF` environment.
  - The standalone taks will be in ANSI standard FORTRAN or C and comform to the OGIP programming standard (OGIP/92-010, Mukai 1992), and use the XPI and/or SAO 'Host Interface' parameter interface routines.
- Use the FITSIO (Pence 1992) and/or FTOOLS (Blackburn 1993) FITS file access routines.

### 5.1 Creation & Updates to CIFs

The following tasks involved with the creation and updating of CIFs are available to the general community:

1. `crcif` ('create CIF'). This routine searches for FITS calibration files containing extensions satisfying standard criteria (see CAL/GEN/92-015, George & Zellar 1993a) within a user-defined directory structure. The required keywords (Section 4) from each extension found are then extracted, and written along with the other necessary keywords (see Section 3.1) into a **new** CIF in a user-defined location.
2. `udcif` ('update CIF'). This routine takes a single calibration file currently in a user-defined directory and checks for extensions satisfying the same standard criteria as `crcif`. The required keywords (Section 4) from each calibration dataset found are then extracted. Following checks that this dataset is not already included within the CIF, the new keywords corresponding to the new dataset are appended along with the other necessary keywords (see Section 3.1) as a new row to a **pre-existing** (user-defined) CIF.

Both the above tasks have an optional input file (`cif.qual`) which supplies information concerning the quality of each calibration dataset to be included in the CIF. These tasks, along with the format and use of `cif.qual` files, are described in more detail in CAL/GEN/92-015 (George & Zellar 1993a).

### 5.2 The (limited) Editing of CIFs

There are only two occasions when a user may need to edit a CIF:

- if a calibration dataset is to be removed from the local caldb, in which case either:
  - the appropriate row can be deleted from the CIF, and the value of the `NAXIS2` keyword reduced by one,
  - or, the quality flag of the calibration dataset (in the **CAL_QUAL** column) can be reset to $Qual = -9$.

The latter is strongly recommended.

- if a the quality flag (in the **CAL_QUAL** column) of a calibration dataset is to be changed.

Both these tasks can be performed using the CALTOOLS task `edtcif`, which also provides the facility to view the contents of a CIF (see Section 5.3.1). Users are **strongly discouraged** from any other editing of CIFs.

Obviously only a very restricted number of users ought to have write priviledge (and hence be able to successfully edit) the primary CIFs of any remote nodes. Individual users who wish to customize their own CIF can, however, copy the system CIF to their own space, make any changes they wish, and reset the appropriate pointer from the `cinitu` file (Section 2.2.1) prior to running any CALTOOLS software.

## 5.3 CIF Inquiries

### 5.3.1 Inquiry Tasks

Since CIFs are in FITS format, a large number of general (publically available) FITS tasks are available to users wishing to manipulate their contents. However, clearly the contents and structure of the original CIF must be left intact if OGIP-supplied calibration s/w is expected to work (but see also Section 5.2).

The following CALTOOLS tasks are of use for CIF inquiries (see CAL/SW/93-004; George, Zellar & Yusaf 1993a, available on-line as pdf and html

versions)

1. `quzcif` ('quiz CIF'). This task finds (using the environment variables discussed in Section 2.2.1) and opens the specified CIF. It then searches through this CIF for entries (rows) corresponding to the calibration data required (*ie.* appropriate for the mission & instrument of interest, and with the desired CCNM$xxxx$ codename).
   If being run in :
   - **'automatic' mode**, the task directly returns the location, filename *etc.* of the 'good' quality calibration dataset appropriate for the time of interest.
     It is anticipated that this will be the normal operating mode.
   - **'querry' mode**, the task lists all available calibrations available, irrespective of their quailty flag or validity start time/date. The user is then prompted for which dataset is desired, and the location, filename *etc.* of this returned.
     This mode is designed specifically for users wishing to investigate the effect on their scientific results of using non-standard, non-recommended calibrations.

2. `edtcif` ('edit CIF'). This task provides an interactive display and (limited) editing facility for CIFs. Users are able to select which rows & columns are to be displayed, and display the full contents of any given row. The results can also be written out to an ASCII file if desired. Users (with write priviledges) are also able to change the value of the followings columns (**only**) of the CIF:

   - CAL_QUAL - the quality flag
   - CAL_DESC - the descriptive string

The following FUTILS tasks are also of use for user inquiry of CIFs (see Pence 1992)

1. `FDUMP` - writes the contents of any FITS file to either the terminal or an ASCII file.
2. `FSELECT` - writes a user-selected subset of the contents of any FITS file to a new FITS file.

#### 5.3.2 Inquiry Routines

The following FORTRAN subroutines available in the calibration subroutine library (`callib`, see CAL/SW/93-005, George, Zellar & Yusaf 1993b, available online in pdf and html versions) are provided to aide s/w inquiry of CIFs:

1. `qzcif`, the primary FORTRAN inquiry subroutine for CIFs. This routine is at the heart of the CALTOOLS task `quzcif` described in Section 5.3.1, and hence is able to perfrom the same tasks as described there.

A number of higher-level individual subroutines will/are also provided within `callib` which given the mission & instrument name, observation *etc.*, finds (using `qzcif`), reads & returns the relevant calibration dataset to the main program. An example is the subroutine `gt_det_eff` which returns the detector efficiency as a function of energy for any given detector[1]. A more detailed desciption of all such members of `callib` is given in CAL/SW/93-005 (George, Zellar & Yusaf 1993b, available online in pdf and html versions).

## 6 INSTALLATION & USE AT REMOTE SITES

Documentation describing the installation, maintenance and use of CIFs:

- **within the OGIP** is provided in CAL/GEN/92-014 (Zellar & George 1993)

[1]Clearly, such routines require that the relevant OGIP caldb calibration files exist within the local caldb. If this is not the case, then a descriptive error flag is returned to the main program.

- **at remote sites** is provided in CAL/GEN/92-015 (George & Zellar 1993a)

As stated above, in order to reduce disk-space and maintenance requirements, it is **strongly recommended that remote users on clustered machines share a single local caldb, and that a single person be responsible for its maintenance.**

## ACKNOWLEDGMENTS

We thank the numerous people, both inside and outside the OGIP, who have contributed ideas and suggestions.

## REFERENCES

Blackburn, K., 1993. In preparation.
Cotton, W.D. & Tody, D., 1992. In preparation.
George, I.M., 1992. *Legacy*, **1**, 56, CAL/GEN/91-001 (available online in pdf and html versions).
George, I.M. & Zellar, R., 1993a. In preparation. (CAL/GEN/92-015)
George, I.M., Zellar, R., & Pence, W., 1993. *OGIP Calibration Memo* CAL/GEN/92-011 (available online in pdf and html versions).
George, I.M., Zellar, R., & Yusaf, R., 1993a. *OGIP Calibration Memo* CAL/SW/93-004 (available online in pdf and html versions).
George, I.M., Zellar, R., & Yusaf, R., 1993b. *OGIP Calibration Memo* CAL/SW/93-005 (available online in pdf and html versions).
Griesen, 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., Griesen, E.W. & Wells, D.C., 1988. *Astron. Astrophys. Suppl.*, **73**, 365.
Mukai, K., 1992. *OGIP Memo* OGIP/92-010.
Pence, W., 1992. *Legacy*, **1**, 14..
Wells, D.C., Griesen, E.W. & Harten, R.H., 1981, *Astron. Astrophys. Suppl.*, **44**, 363.
Zellar, R. & George, I.M., 1992. In preparation. (CAL/GEN/92-014).