Technical Instrument Report OSG-CAL-97-02 (updated 7/1/98)

Delivering Calibration Reference Files

C. Cox, C Tullos
April 1, 1997 (Updated 7/1/98)

ABSTRACT
The redesigned Calibration Database System requires a new delivery protocol. This report describes the software and procedures to be used by the deliverer and installer.

1. Introduction

When calibration reference files are delivered, two main things happen. The files are copied to three locations and their existence is recorded in three databases. The files are first saved to one of several disks on the Unix science network. They also get placed in the DADS archive and on disks owned by OPUS. Pointers to these files based on instrument modes and applicability dates are placed in the Calibration Data Base System (CDBS) and in OPUS's database. A database parallel to CDBS is also maintained in DADS. The main function of the databases is to allow selection of the correct reference files based on an observation's instrument configuration and date, both for the calibration pipeline and for off-line recalibration. A second function is to make information about the history and pedigree of each file permanently available

2. Instrument Team Procedures

Setting up

The software used in preparing reference files for the calibration database is in the directory `/data/cdbs1/tools/bin` This path should be declared in your `.setenv` file in your home directory.

Delivery steps

1. Place all the files belonging to a single delivery in their own directory. This can include different types of calibration files, but all must refer to a single instrument. Cross-instrument files are considered to belong to a single extra instrument.

2. SPECIAL CASE: (Synphot tables) -- These tables need to have certain keywords and values in the headers (see the iraf task *tupar* for reading/editing headers). The keywords are as follows:

   DBTABLE t CRTHROUGHPUT

   INSTRUME t acs (use lower case)

   COMPNAME t acs_hrc_m12 (use lower case)

   DESCRIP t text field (description of the data)

   HISTORY t text field

   COMMENT t text field

   USEAFTER t Jun 25 1998 12:12:12 AM

   PEDIGREE t ground,model,inflight,dummy (pick one only)

Note: The format of the USEAFTER keyword needs to be in the form (Mmm dd yyyy hh:mm:ss AM/PM) and the hh:mm:ss am/pm part is optional. Also, the value for INSTRUME needs to be in lower case.

3. Run *fitsverify* on all FITS files. This checks that the files are compliant with the FITS standard. DADS will repeat this test and will not archive any files that fail. This task can be run without any special setup from the tib cluster with the command `fitsverify *.fits`

4. Run *certify* on all the files. This ensures adherence to the ICD-47[1] requirements. The command can be `certify *`

   If any fail, patch them up until everything passes. If you do not understand why the files are not in compliance, please refer to ICD-47. If any of the error messages are unclear, contact CDBS personnel for assistance.

5. Run *mkload* on all files. The command can be `mkload *` to do them all at once. A load file with extension .lod will be generated for each calibration image and table.

6. Edit the load files.

   a) To indicate whether they are intended for OPUS or not, set OPUS_FLAG to Y or N.

   b) Define the level of change with respect to earlier matching deliveries by setting

      CHANGE_LEVEL to SEVERE, MODERATE, or TRIVIAL.

CHANGE_LEVEL should be SEVERE for initial deliveries.

In rare cases you may want to enter your own comparison file name, but the system will supply the name of the most recently entered file that matches the instrument mode and useafter date.

c) Supply a COMMENT for each file. This can be a lengthy description. Use as many lines as necessary. You can also enter a COMMENT for each row of a table. *Please note the importance of this comment field!! This will be the only comment field to be written to the database and seen by starview. The various other fields (comment,history,descrip) are kept with the data (headers), but not accessible by starview.*

7. Run *certify* and *check_load* on the load files. The commands can be `certify *.lod` and `check_load *.lod` If the load file fails *certify* at this point either it will be some simple matter like misspelling “severe” or a more serious problem such as an erroneous template file that would have to be investigated by CDBS Operations.

8. If *certify* shows values of “ANY” or “-1” and the calibration file is not a table, the load file has to be expanded. Do not run *expload* on tables. If you want to expand a table, run the iraf task *texpand* on the original table and deliver it in the usual way.

   a) Run the task *expload*. The form of the command is

```
expload <input_load_file> <output_load_file> <rule_file>
```

   where the angle brackets indicate this not literally what you type but you should substitute actual file names. An example is:

```
expload band2_super_pfl.lod super.lod stis.rule
```

The output load file must have the same rootname as the related calibration file. You can achieve this by renaming the output load file when you have finished certifying. Alternatively you can give the output file the name of the input file. This overwrites your input load file. This is not a serious risk because in case of difficulties you can always regenerate the original load file although you will have to repeat steps 5a, b and c. If your comment is lengthy, you may wish to save it and paste it into the new load file.

   b) Remove the unexpanded load file if it is still present.

9. Rename all files to their unique time-stamp name with the command
   `uniqname *`
   The data files associated with the load files will be renamed in step. **This will replace your original data and load files so if you need to keep the files with their original names be sure you are working on copies.**
   By redirecting the STDOUT output to a file, you can retain the association between old and new names. A history of the name change *is* written automatically by

uniqname in the header of the file. Synphot deliveries require special handling by CDBS Operations. The throughput table names do not change (root), but the numerical increment does change (_002 -> _003) and uniqname knows how to do this.

10. Complete the email delivery form and mail it to CDBS Operations (Tullos@stsci.edu). We take it from there and will send you a report when installation is complete. A sample email delivery form is shown.

    date:

    by:

    instrument:

    file_type(s):

    directory where data is found:

    data description or reason for delivery:

    opus injest date:

    opus signoff:

    *note: please include a listing with name/size of what you are delivering

Supporting Data

The program *certify* checks that certain keywords are present and have legal values. The data to compare with reside in a set of "template" files maintained in the directory `/data/cdbs1/tools/data.` They have names like `foc_bkg.tpn` and `foc_bkg_ld.tpn.` The ones containing `_ld` are for the load files. Although these are maintained by CDBS Operations they will not be altered without being reviewed by the instrument team. Similarly *expload* uses a file such as `stis.rule` giving the expansion rules. The rules should be supplied or reviewed by instrument teams. These files are in the same format as is used by the iraf task *texpand*. One file will be maintained for each instrument in the same data directory. This procedure is most commonly used to permit dummy reference files to refer to multiple instrument modes before in-flight calibration data are available. Rule files may not be needed for the mature instruments.

3. Installation in CDBS

The delivered calibration and load files are checked again and when all problems are resolved, SQL commands are generated to load the data into CDBS and OPUS. The CDBS SQL is run and the matching OPUS SQL passed on with the files for OPUS to run. Copies of all calibration files are sent to the data archive and recent ones are maintained in directories on tib. The detailed steps follow. Naturally, each step is only undertaken following successful completion of the previous one. Although each step is described individually here, we now have a script *deliver_cdbs* which runs all of them, breaking out only when an error condition is encountered. In this case it may be necessary to correct the error and perform the remaining steps individually.

Before running any of the software, issue the command

```
source /data/cdbs1/tools/util/defines.csh
```

This will set the correct environment variables to access the CDBS database.

Detailed steps

1. Create a new working directory under `/data/cdbs1/work` Copy all files here from the delivery directory. (At this point *deliver_cdbs* can be started and normally replaces steps 2 through 9.)
2. Run *certify* on all files
3. Run *check_load* on all load files. Only some of the load file values are intended to be edited. Items such as the USEAFTER_DATE are read from file headers and should not be edited. This check looks for any discrepancies.
4. Build the SQL to populate CDBS with the command `cdbs_sql_gen *.lod` This will create the SQL command script
5. Populate CDBS with the command `run_cdbs_sql *sql`
6. Run *check_cdbs* and examine the output which will be in a file named check_cdbs_nnnnn.out where nnnnn is the current delivery number. Decide whether the delivery can proceed. There may be messages concerning installation and archiving of previous deliveries which should be investigated, but are not a problem for the current delivery.
7. Normally you will proceed with the command `update_ga_date`. This updates the value of general_availability_date and fixes the delivery in the calibration data base. However, if *check_cdbs* has revealed serious problems do not run this. Instead, issue the command `delete_delivery nnnnn` and resolve the problem before attempting the delivery again. CDBS will contain no record of the delivery attempt. Once *update_ga_date* has been run, *delete_delivery* has no effect.
8. Run *opus_sql_gen*. This makes up the SQL commands for loading the OPUS database and outputs the file opus_nnnnn_x.sql This command file is not run by CDBS but passed on to OPUS. x is an instrument code letter.

9. Run *opus_catalog*. This generates a list of files to be delivered and those to be superseded in opus_nnnnn_xcat. OPUS will actually delete the superseded files from the active collection. This is the last step incorporated in *deliver_cdbs*.

10. For all instruments except STIS and NICMOS, files will be delivered as STSDAS images and tables. In this case run the iraf script *loopfits* which will convert all the files to FITS.

11. Run *fitsverify* on all the FITS files. The command is simply
    `fitsverify *.fits`

12. Copy the renamed FITS files to the directory `/data/cdbs1/opus_release` Next copy the OPUS SQL and catalog files into the release directory. It is important to copy the FITS files before the other two files. Send a mail message to Mike Swam (mswam) to let OPUS know to pick up the delivery. Also take the delivery form to the OPUS operations room keeping a photocopy until the signed copy is returned.

13. Prepare the data for DADS by copying the FITS files to the directory `/data/cdbs1/dads_release` Move to the same directory and issue the command
    `cdbs_make_request -a cdb` or `cdbs_make_request -a cbt` according to whether you are delivering images or tables.

14. Copy all calibration files, not the load files, to the CDBS on-line archive. This means copying to specific directories according to the instrument and whether the files are images or tables. The placement is as follows.

	Image	**Table**
NICMOS	/data/cdbs5/nref	/data/cdbs5/ntab
STIS	/data/cdbs4/oref	/data/cdbs4/otab
WFPC2	/data/cdbs3/uref	(none)
FOC	/data/cdbs2/xref	/data/cdbs2/xtab
FOS	/data/cdbs2/yref	/data/cdbs2/ytab
HRS	/data/cdbs2/zref	/data/cdbs2/ztab
MULTI	(none)	/data/cdbs2/mtab
SYNPHOT	(none)	/data/cdbs2/comp

    For all instruments other than STIS and NICMOS, the STSDAS files are used for this purpose. (For STIS and NICMOS, only FITS files exist.) Delete all the data files from the working directory.

15. The final step is to run the program *cdbs_report* once the delivery has been accepted by OPUS and DADS. The report states the dates and times of OPUS

installation and archiving for each file. A copy of the output is to be e-mailed to the person delivering the files. At present we know when to run this by the mail messages automatically generated by the OPUS installation and archiving steps. This will soon become a totally automatic step including the mailing to interested people.

Special considerations for synphot deliveries

Synphot or throughput deliveries require a slight variation in procedure, actually causing two deliveries. There are normally several throughput tables with the extension `.tab` and possibly a graph table with the extension `.tmg` The `.tab` tables do get renamed, although the only change is in the numerical incrementation. The steps are:

*** assume that uniqname has already been run!!*

1. Separate graph table from thruput tables and put into another directory.

2. Copy thruput tables into their appropriate online directories.

3. Run deliver_cdbs on thruput tables and associated load files.

4. In separate directory, run mkcomptab (mkcomptab tmp.tmc)

5. uniqname *.tmc

6. certify *.tmc

7. mkload *.tmc

8. Edit load file (put in `change_level, pedigree, comments, and opus_flag`).

9. run certify and check_load on load files.

10. Include graph table (.tmg) with master component table (.tmc). Copy both to /data/cdbs2/mtab.

11. deliver_cdbs to install comp and graph tables.

12. Continue with regular delivery procedures.

13. Supporting Software

Several programs are supplied to facilitate the delivery procedure. Only the first four listed will be used by those delivering files. The remainder are for CDBS operations.

certify checks that required keywords are present in the headers of calibration reference files and contain legal values. This does not directly test whether the files will perform correctly in the calibration software but only that they can be installed in CDBS with all necessary pointers and history information.

mkload generates a file, which we call a load file, from each delivered reference file. The load file is a simple text file containing copies of the essential delivery keywords. Some manual editing of the load file is required to enter information about the calibration file that cannot be automatically assigned. This is where descriptive comments must be entered.

expload is used when a single reference file supports multiple instrument modes. A typical use of this facility is to supply a dummy reference file before real data are available. The program expands the load file. adding rows to describe this situation. A specially formatted rules file is needed which describes precisely how the expansion is to be done.

uniqname changes the names of the data and load files according to the current time and date. This renaming ensures that no two files with the same name are installed in the databases. The instrument code is embedded in the name together with the calibration file type. The linkage between load files and data files is retained

check_load examines the load file and ensures that entries that are generated from the header do not get changed. There are several entries that have to be edited but mode parameters and useafter dates must match the values in the delivered files.

cdbs_sql_gen reads the load files and generates SQL commands which load the reference information into database relations.

check_cdbs runs various checks on the calibration database, referring principally to the most recent delivery. One check is to prohibit two files with the same configuration from being in the same delivery set. Also, if previous deliveries have not been installed or archived, warnings are issued.

opus_sql_gen takes its information from CDBS for the current delivery and prepares SQL commands to update the OPUS database.

update_ga _date sets the general availability date for each delivery. This implies that all files in the delivery are ready for OPUS and DADS to accept. Once this date-time stamp is set, the references to these files will never be removed from CDBS.

opus_catalog generates a list of the files in the current delivery plus a list of previously delivered files that are now superseded and may now be deleted from the OPUS active set of files.

Documentation for most of these programs is supplied in `/data/cdbs1/tools/doc`. The documents are in the form of LaTeX files which can either be read as text files containing many annoying special characters, or printed out using ptex.

References:

[1] Post Observation Data Processing System to Calibration Data Base System Interface Control Document (ICD-47) SCI88075B