



Operated for NASA by AURA

Technical Instrument Report CDBS 2005-01

Assessment and Delivery of Reference Files

R. I. Diaz-Miller
rmiller@stsci.edu

May 17, 2005

Abstract

This TIR defines the standard procedures for the delivery of calibration pipeline and SYNPHOT reference files. It clearly establishes the instrument and OTA teams' responsibilities and provides guidelines for testing and verification of reference files.

Introduction

The generation of calibration and SYNPHOT reference files is currently the responsibility of the instrument and OTA teams at STScI. As part of this process, standard testing and assessment of the files need to be performed before they are “delivered” officially to the INS/CDBS (INS Calibration Data Base System) Group for their ingest into the CDBS, DADS, and OPUS databases. In order for the reference files to be used by the OTFR pipeline, those files have to be ingested in the OPUS database. Their ingestion into the Data Archive Distribution Services (DADS) database allows users to retrieve them from the archive and mirror sites. The main function of the CDBS database is to allow selection of the correct reference files for a specific set of observations. The selection is based on data file keyword values and the selection criteria outlined in ICD-47, located at <http://www.stsci.edu/instruments/observatory/cdbS/documents/icd-47RevE.pdf>.

The checking and quality assessment of reference files will be performed by the person creating the files, while the official delivery of the files to the databases is handled by the INS/CDBS Group. Checking and quality assessment of reference files is required for the following reasons: to ensure correct keyword syntax, file structure and format; to ensure that the files will work properly in the pipeline calibration software; to avoid propagation of erroneous files; and to ensure proper documentation of the reference files. Accordingly, we define in this document the standard procedures for checking and assessing the quality of reference files. The procedures outlined here should follow the creation of the reference files, which are assumed to be in ICD-47 format. We first provide an outline of the steps to be performed, followed by a more detailed description. If errors are encountered, these must be resolved before proceeding to the next step. Unresolved

problems with the CDBS tasks should be referred to the INS/CDBS Group (cdbbs@stsci.edu). The detailed steps and procedures for ingesting reference files into the databases are described in TIR CDBS 2005-02.

Summary: Preparation and Assessment of Spectrographs Reference Files

1. Update all the reference files header properly (including header keywords and history lines).
2. Rename working copies of the files to “waiver” FITS format.
3. Run the CDBS `certify` tool on the standard FITS or header GEIS files.
4. Perform a detailed quality assessment of the data in the reference files.
5. Test the file using `CALXXX` or `SYNPHOT` and `ETC`.
6. Fill out the delivery template and notify the INS/CDBS group that the files are ready for delivery.

Detailed description of the preparation steps

1. Update all the reference files header properly (including header keywords and history lines)

Once the reference files have been created, it is necessary to update the primary headers with the proper useafter date, pedigree of the file, description, comment keyword, and history lines. (Header keywords that are instrument specific, like `DETECTOR` or `FILENAME`, should be filled in compliance with the formatting outlined in ICD-47 document.) These header keywords are crucial for the correct selection and documentation of the reference files and can be added using the IRAF task `hedit`. The relevant header keywords are: `USEAFTER`, `PEDIGREE`, `DESCRIP`, and `COMMENT`. For `SYNPHOT` data files (also known as throughput tables) three extra keywords have to be present: `INSTRUME`, `COMPNAME`, and `DBTABLE`. The `USEAFTER` and `PEDIGREE` (and `INSTRUME` and `DBTABLE` for `SYNPHOT` data files) keywords most likely will exist already in the header; `DESCRIP` and `COMMENT` (and `COMPNAME` for `SYNPHOT` data files) may need to be “added”. This can be done by setting the `add` parameter equal to `yes` in `hedit`. Note that the information contained in the `USEAFTER` date keyword is used differently by the pipeline and `SYNPHOT` software. In order to avoid confusion, each of these cases is discussed independently. Also, if the reference file primary header has information that is not relevant to the reference file itself (e.g., calibration keywords, observation parameters, calibration switches, etc., of one of the datasets used to create the file), erase them before delivering the file. Make the primary header short, concise, and clear.

USEAFTER (*in Pipeline Reference Files*)

This is the most crucial of all the header keywords in a reference file. It indicates the date and time after which this file should be used in the calibration of science data. The format for this keyword is *month dd yyyy [hh : mm : ss]*, where the time entry is optional. For the month, any valid abbreviation is acceptable (or write out the whole word). The time should be set equal to 00:00:00 for most of the reference files, which is the same as not using a time entry. Exceptions to this case are for those files, like STIS or ACS bias and dark reference files, that should not be used across anneal occurrences and therefore the time is set equal to the anneal time. For example, a reference file that has a useafter date of “*Jan 26 2005 05 : 23 : 53*”, should not be used to calibrate data taken BEFORE that date and hour. Furthermore, although most reference files are “back date” to launch in order to be used for all data, there are cases where more than one file of the same type and for the same mode exist, but with different USEAFTER dates. Special care should be taken in deciding the useafter date for those cases, and make sure that they correctly will supersede outdated reference files or cover the specific date ranges. As a rule of thumb, for those new reference files that will replace an old reference file, the exact USEAFTER date and time of the old reference file should be used for the new reference file.

USEAFTER (*for SYNPHOT Data Files*)

The USEAFTER keyword in SYNPHOT data files (also known as throughput tables) is also important for the selection of the correct data files. In this case, however, it is used in a different way than the USEAFTER date keyword in the pipeline reference files. In particular, the SYNPHOT software itself does not care about this keyword and it is used only by the CDBS database to determine when the files were created. This information in turn is used by one of the CDBS tools, the MKCOMPTAB task, that creates the Master Component Table (MC table), or “lookup table”. For a particular SYNPHOT data file type, the MKCOMPTAB task looks into the CDBS database and identifies the files with the most recent USEAFTER dates. The most recent files then are used to populate the MC table for each of the SYNPHOT components. Therefore, the USEAFTER date in the SYNPHOT data files should be populated with the “date when the files were created.” The format for this keyword is “*Month dd yyyy [hh : mm : ss]*”, where the time entry is optional.

Note that if the CDBS tool MKCOMPTAB finds several versions of a data file of the same type with the same USEAFTER date, it will use all of them to populate the MC table and therefore there will be repeated entries for the same SYNPHOT component. In this case, the order in which those files appear is important and SYNPHOT will use only the last of those data files listed in the MC table.

PEDIGREE

The header keyword PEDIGREE describes the type of data used to create the reference file: INFLIGHT, GROUND or DUMMY. For the case of PEDIGREE = INFLIGHT, the range of dates of the data used to create the file should be included. A typical inflight pedigree looks something like this:

‘‘INFLIGHT 21/01/2000 26/01/2000’’, with the dates in *dd/mm/yyyy* format (note the four digit year). If the data in a reference file have more than one kind of pedigree (e.g., there are rows with pedigree **INFLIGHT** and **GROUND**), set the header keyword **PEDIGREE** to the one occurring more in the reference file, as only one kind of pedigree can be used in the header.

DESCRIP

The **DESCRIP** (Description) keyword should be a concise sentence; one that makes it easy for users to decipher the type of reference file. For example: **DESCRIP** = ‘‘**Reference superdark created from proposal 7276**’’, provides a simple description of the file. More complete information will be provided in the **HISTORY** lines.

COMMENT

In the **COMMENT** keyword, the names of the creators of the reference file should be entered; e.g., ‘‘Reference file was created by R. Diaz-Miller and P. Goudfrooij.;;

DBTABLE (for SYNPHOT data files only)

The **DBTABLE** keyword should be set to ‘‘**CRTHROUGHPUT**’’ for all SYNPHOT throughput files.

COMPNAME (for SYNPHOT data files only)

The **COMPNAME** keyword should have the SYNPHOT name of the component to which the delivered reference file applies. This is usually the first part of the name of the file you are delivering. For example, if you are delivering the data file ‘‘stis_a2d8_003_syn.fits’’, the **COMPNAME** should be set to ‘‘stis_a2d8’’. Check previously delivered files to make sure that you are using the right value.

INSTRUME (for SYNPHOT data files only)

The **INSTRUME** (instrument) keyword should have the instrument to which the SYNPHOT data files apply. The instrument name should be in lowercase.

HISTORY

The HISTORY lines can be added to the file header in many different ways, here we will mention only some of them.

One way to add the history section is via the IRAF task called `stfhistory` in the `stdas.toolbox.headers` package. This task loads an ASCII file, given in the parameter “text,” into your reference file header as history lines, pre-appending the word HISTORY to each line. For example:

```
PACKAGE = headers
TASK = stfhistory

input = reference_file.fits[0] Image template
text = @historyFile.ascii History text
(verbose= yes) Verbose message switch
(mode = al)
```

will input the text contained in the file ‘‘historyFile.ascii’’ as HISTORY lines in the header [0] of the reference file `reference_file.fits`. Make sure that in this ASCII file each line has 62 characters or less, otherwise the `stfhistory` task will break the text as needed and you will end up with a HISTORY section that has rows of different length.

The following information is required in the HISTORY lines:

1. A complete but concise description of the process used to create the reference file. Be specific –note what software was used and if it is available to the GOs or only internally.
2. Include in the description any pertinent numbers that others may find useful, e.g.- how many crsplits or repeat observations were involved to create the image, what was the total exposure time, the kind of software used to create the reference file, or the value above which hot pixels were updated.
3. When possible, include relevant ISR/TIR numbers.
4. A listing of what raw datasets went into the creation of the reference file.
5. No names of people involved in the creation of the reference file should be included in the HISTORY lines, these names only should be mentioned in the COMMENT keyword. This is because the COMMENT keyword does not show up in StarView.

Keep in mind that history lines are very important because they not only document what we did to create the reference file, but they serve to inform general observers who use the file as well. In some cases it can be useful to look at the headers of previously delivered reference files to see what was placed in their history lines. However, in those cases where a file has changed appreciably since the last delivery (i.e., if most of the rows have changed), rather than repeat history lines and produce a lengthy output that is hard to follow, simply reference the last

appropriate reference file and TIR/ISR number in the history section of the new file. An example history file looks something like this:

```
Created on July 26, 1999, using the
cl script "weekdark", which is available in the
stis package within STSDAS.
```

This superdark image is a combination of 2 images. The first is a "baseline dark" image which is a (typically) monthly average of (cosmic-ray-rejected) dark files (in this case an average of 38 darks). The second image is made locally within the "weekdark" script from 14 darks that are taken from Proposals 7948/7949/8408/8437 "Dark and Bias Monitor". These gain 1 darks are combined together (cosmic rays are rejected in the combination) using calstis, and normalized to a dark time of 1 second. After that, hot pixels in that normalized dark are updated into the baseline dark. These hot pixels have a value higher than (baseline dark current + 5 sigma of the new dark). Hot pixels have been updated above a level of 0.01723591 electrons/second for this particular dark. The pixels hotter than 0.1 electron/sec in this dark are being assigned a DQ value of 16. Previously hot pixels that have fully annealed out between the observing dates of the baseline and the new dark are being assigned a value equal to that in a median-filtered (kernel = 2x2 pixels) version of the baseline dark.

The following input dark files were used:

```
o4wi7agxq
o4wi7bh5q
o4wi7cnyq
o4wi7doiq
o4wi7ew1q
```

If you should happen to make a mistake in your history lines and only notice it after you have added it to your primary header with `stfhistory`, you can "undo" what you did with a task called `ehheader` or `tupar`. The IRAF `stsdas.toolbox.eheader` task will allow you to edit the primary image header using an interactive text editor. (It might work only with "vi".) After editing is complete, the changes are saved back to the original file. For example, to erase lines in "vi" go to the line you want to erase using the arrow keys and then type "dd". To erase a word, type "wd"; to erase one character, type "x". To insert text, type "i" and then start inserting your text; to escape insertion mode, use the < ESC > key. To save your changes, type "< ESC >: wq < return >".

This last command will take you out of the “vi” editor. Notice that the `eheader` task also can be used to add history lines. In this case you should remember to add the word “`HISTORY` ” (leave four spaces after this word) at the beginning of the line. To erase lines with `tupar` simply `tupar` the file’s primary header: `tupar reference_file.fits[0]`. At the colon prompt type “`l`” to list the header. Once you see what line numbers you would like to delete type: “`d #`” where `#` is the line number. Typing “`e`” will exit and save the file; typing “`q`” will exit without saving the changes you just made.

Special considerations for “waiver” FITS files

In the case of “waiver” FITS files (e.g., WFPC2 dark and bias), you have to make sure that the headers are FITS compliant. For this you might have to replace tabs by spaces and to check for single quotations.

2. Rename files to contain no capital letters.

CDBS cannot process files with names containing capital letters. Such files should be renamed to contain only lower case letters. If you are delivering a large number of files with capital letters in their names, the following command can be used to do the renaming automatically:

```
decap filename
```

3. Verify that the files are in standard or “waiver” FITS format.

Fits reference files that are not in standard or ”waiver” FITS format cannot be ingested into the databases. To verify that reference files are in standard FITS format, run the `fitsverify` tool on the files:

```
fitsverify filename.fits
```

Wildcards may be used for filenames, e.g., `filename.fits` in this example can be replaced by `*.fits`. A sample output from the tool looks like this:

```
=====
FITS Verification for file: lbq1211ao_bia.fits
=====
Summary contents of FITS file: lbq1211ao_bia.fits
0: Primary Array ( SHORT )
0 bytes, 108 header lines, 3 FITS blocks
1: Image Extension ( FLOAT ) [IMAGE,SCI,1] 2 dims [1024,1024]
```

```
4194304 bytes, 36 header lines, 1458 FITS blocks
2: Image Extension ( FLOAT ) [IMAGE,ERR,1] 2 dims [1024,10w24]
4194304 bytes, 36 header lines, 1458 FITS blocks
3: Image Extension ( SHORT ) [IMAGE,DQ,1] 2 dims [1024,1024]
2097152 bytes, 36 header lines, 730 FITS blocks
No special records.
```

```
=====
No problems were encountered.
```

Examples of problems encountered with the files in FITS verification include:

- extra spaces in keyword fields
- incorrect format for DATE keyword field (Dec. 18, 2000 instead of 18/12/00)
- missing PCOUNT and GCOUNT keywords in extension headers

Note that this command checks for standard FITS header keywords and does not report any missing instrument-dependent header keywords; i.e., those that give the information necessary to define the observing mode to which these files apply and that should be present in all reference files. A complete list of the standard FITS header keywords can be found in ICD-47.

4. Run the CDBS certify tool on the standard FITS or header GEIS files

The CDBS `certify` tool performs additional checking on the syntax and keyword values in the standard FITS files or the header GEIS data files (currently for WFPC2 only), ensuring adherence to ICD-47 specifications for each type of reference file. Instrument specific header keywords and columns (in a table file) that are necessary for the correct selection of a reference file will be checked. This tool is run by typing a UNIX shell line (Note that most CDBS UNIX scripts can also be accessed through IRAF in the `stlocal.cdbutil` package):

```
certify filename
```

Wildcards may be used for filenames, e.g., `*.fits` or `*.*h`. More detailed documentation on the `certify` task is available in postscript format on the CDBS web page (<http://www.stsci.edu/hst/observatory/cdb/util/documents/>). The `certify` tool does not check all the keyword syntax and values in the reference file, but only those that are used specifically in CDBS, OPUS, and DADS systems for selecting and tracking the reference files. A complete list of the standard instrument-dependant header keywords can be found in ICD-47. The required keywords are accessed by `certify` via CDBS template files. There is one template file for each reference file type and these are located in the CDBS working areas of the science cluster and the `small.stsci.edu` domain. A sample of the template file for the STIS PHT reference file looks like this:

```
# Template file used by certify to check reference files
```

```

# Some fields may be abbreviated to their first character:
#
# keytype = (Header|Group|Column)
# datatype = (Integer|Real|Logical|Double|Character)
# presence = (Optional|Required)
#
# NAME KEYTYPE DATATYPE PRESENCE VALUES
#-----
INSTRUME H C R STIS
FILETYPE H C R "PHOTOMETRIC CONVERSION TABLE"
DETECTOR H C R CCD,NUV-MAMA,FUV-MAMA
OBSTYPE H C R IMAGING,SPECTROSCOPIC
OPT_ELEM C C R G140L,G140M,E140M,E140H,G230L,\
G230M,E230M,E230H,PRISM,G230LB,G230MB,G430L,G430M,G750L,G750M,\
MIRCUV,MIRFUV,MIRNUV,MIRVIS,X140H,X140M,X230H,X230M,N/A
CENWAVE H I R
1173,1200,1218,1222,1234,1271,1272,\
1307,1321,1343,1371,1380,1387,1400,1416,1420,1425,1453,1470,1489,\
1518,1526,1540,1550,1562,1567,1575,1598,1616,1640,1665,1687,1713,1714,\
1763,1769,1813,1851,1854,1863,1884,1913,1933,1963,1978,1995,2013,\
2014,2063,2095,2113,2124,2125,2135,2163,2176,2213,2257,2263,2269,\
2276,2313,2338,2363,2375,2376,2413,2415,2416,2419,2463,2499,2513,\
2557,2561,2563,2579,2600,2613,2659,2663,2697,2707,2713,2739,2762,\
2794,2800,2812,2818,2828,2836,2862,2898,2912,2962,2976,2977,3012,\
3055,3115,3165,3305,3423,3680,3843,3936,4194,4300,4451,4706,4781,\
4961,5093,5216,5471,5734,6094,6252,6581,6768,7283,7751,7795,8311,\
8561,8825,8975,9286,9336,9806,9851,10363,\
1232,1269,1305,1341,1378,1414,1451,1487,1523,1560,1587,1760,\
2010,2261,2511,2760,3010,1975,2703,-1,-999
USEAFTER H C R &SYBDATE
PEDIGREE C C R &PEDIGREE
DESCRIP C C R w

```

A sample of the output of certify for a file that has a problem would be:

```

== Checking mama2_PFL.fits ==
Could not match keywords in header (mama2_PFL.fits)
Cannot determine reference file type (mama2_PFL.fits)

```

If you encounter a problem at this stage, first check to see if there are any obvious problems with the file header keywords or keyword values. The list of required and valid values for the header keywords can be found as well in ICD-47. If this check does not reveal the source of the error, contact the INS/CDBS Group.

5. Perform a detailed quality assessment of the data in the reference files.

The integrity of the calibration reference files is critical for the production of high-quality science results. It is therefore important that before delivering them to the pipeline or SYNPHOT database, each Instrument group thoroughly test and validate them to the best of their ability. In addition to checking for proper structure and syntax, checking of the data themselves is essential.

For those cases where automatic checking has not been implemented, there are many IRAF tasks that can be used to examine files. For headers: `eheader`, `imhead`, `hedit`, `tlcol`; for data: `listpix`, `mstat`, `tread`, `tstat`, `display`. Consistency is recommended strongly for the test and integrity checking of the reference files. Some of the questions that might be answered before delivering the files could be: Do the new files look like the old ones? If not, why? Have the column PEDIGREE values been updated according to the changes made in the rows? Do the comments in the updated row have the correct information? Are the error columns or error extension consistent with the data? Keep careful notes on how you determined that the new reference file was good. For image reference files, it is good to run the IRAF task, `imstat` on the file and determine if the values are reasonable in all extensions: science, error and data quality. Often plotting or displaying the file will reveal obvious problems that otherwise might be missed.

6. Test the reference files using CALXXX or SYNPHOT and ETC.

CALXXX must be used to calibrate actual data with the reference files being readied for delivery. The highest level check should verify that the appropriate stages of CALXXX do not crash and that the calibrated data makes sense. Make sure that the proper CALXXX switches are set to `PERFORM` so the files being tested actually are used. Real datasets should be calibrated with both the new reference file and the old reference file, and the differences understood and thoroughly documented. Even in the case of simple changes to the file, the deliverer should make sure that CALXXX runs using these files.

SYNPHOT data files, on the other hand, have to be tested using the SYNPHOT and ETC software. The ETC software relies heavily on the SYNPHOT package to predict counts rates and S/N values via calls to SYNPHOT tasks. However, the results that the ETC produces also depend on calculations performed by the ETC software itself. In order to maintain the integrity of both systems, changes made in SYNPHOT data files have to be tested against the ETC software. Currently, we are working with the ETC/APT project to create a testing environment where the Instruments and OTA teams can test new SYNPHOT files. In this environment, the deliverer will identify those ETC modes that are affected by the changes made to the SYNPHOT data file and will make sure that the differences in the outputs of the external ETC and testing environment ETCs are the expected ones. In order to verify consistently the integrity of the system, we recommend to maintain an ETC test suite that can be used in those cases where the new SYNPHOT reference file will affect all the modes covered by the ETCs. This document will be updated when this environment is in place.

Testing the file in SYNPHOT

This can be accomplished by creating a dummy “Master Component Table” (MCT) that points

to the data files you are delivering. For this, first make a copy of the most current MCT table in your working directory. Rename this file and edit the row for the COMPONENT associated with the data file you are delivering. If there is more than one row for the same COMPONENT in the table, make sure you edit only the last one. Change the FILENAME column of this row to the data file being delivered. You can edit this file via the IRAF task `tedit`. Now modify the parameter “`cmptbl`” in the `stdas.hst_calib.synphot.refdata` task with the name of the dummy TMC file. If you are delivering a “Master Graph Table” (TMG) together with your new data file, you also have to modify the parameter “`grtbl`” of the `refdata` task. Now run the SYNPHOT task that will use this data file. Make sure that you used the correct SYNPHOT data files by running the task `stdas.hst_calib.synphot.showfiles`.

Testing the file with ETC

To be implemented.

7. Fill out and e-mail the delivery template.

When all of the above steps have been completed with no errors, the reference files are ready for delivery. Fill out the delivery template (shown below) and send it to the INS/CDBS Group (`cdbs@stsci.edu`). This template, besides helping the INS/CDBS Group to deliver correctly the files to the Data Management Systems, can be used as a check list for all the steps indicated above. Make sure that the files and directory where the files are located are accessible by anybody. The delivery template also can be found in the INS/CDBS web page under Data Delivery Procedures (http://www.stsci.edu/hst/observatory/cdb/delivery/delivery_procedures.html) and contains the following information:

- 1- Name of deliverer:
(other e-mail addresses)
- 2- Date of delivery:
- 3- Instrument:
- 4- Type of file (bias,pht,etc.):
- 5- History section in header [0] complete? (yes/no):
- 6- USEAFTER, PEDIGREE, DESCRIP, and COMMENT have been checked? (yes/no)
- 7- CDBS Verification complete? (*fitsverify, certify,etc.*):
- 8- Should these files be ingested in the OPUS, DADS and CDBS databases? (if not indicate clearly which ones)
- 9- Files run through CALXXX or SYNPHOT? (yes/no):
- 10- Does it replace an old reference file? (yes/no):
- 10a- If yes, which one?
- 11- What is the level of change of the file? (e.g. compared to old file it could be: SEVERE, MODERATE, TRIVIAL, 1%, 5% etc.):
- 12- Description of how the files were "tested" for correctness:
- 13- Disk location and name of files:

In this template, in the field “(other e-mail addresses)” you should provide all the e-mail addresses

of people or mailing lists that should be notified of any problem or success of the delivery. However, you can leave this field blank if these e-mail addresses are in the Cc field of the e-mail you will be sending to the INS/CDBS Group with this form. The field “Date of delivery” should have the date when you deliver the file to the INS/CDBS Group. In question number five, regarding the completion of the history section, you should put “yes” only when you have made sure that the history section of the file describes the changes made to the file or explains how the values in the file were derived. If you deliver the file with answer “no”, we might have to contact you again to make sure this section gets filled properly. Although most of the reference files usually are delivered to the OPUS, DADS, and CDBS databases, there could be cases when teams choose not to do so; indicate this in question number eight. In question ten, we want to confirm that any delivered reference files correctly replace, when applicable, other files currently in the system. It would be helpful for us to know the name of the file that will be replaced. In order to properly populate the databases, we also need to know the level of change of the delivered files compared to old files or if it is a new file. This information should be given in question eleven. If the file is a new reference file or it covers a range of dates not covered previously, the level of change should be set to SEVERE or 100%. Another case when it should be considered as SEVERE is when the change warrants recalibration of the data. Note that in the case of tables the changes might apply to some rows only. Please let us know which rows were affected by this level of change as we want to identify which datasets are affected. If the changes will have a moderate impact in the results of the calibration, or just make them better but do not warrant recalibration, then the change is MODERATE. Also you could specify the percent change; for example, for the throughputs or images. If the changes made to the file just were to correct minor mistakes and they will not impact or will have negligible impact on the calibrated products, the change level is considered TRIVIAL. We also need information on how the files were “tested” for correctness. In question twelve, briefly describe which kind of test you performed to assess the scientific correctness of the files. For example: “Verified no change for first order modes when compared to previously ingested reference file. Calstis ignores this file for echelle and imaging modes.” or “All extensions examined with imstat to count number and size of hot pixels. Run through calstis with data sets from test suite to ensure that changes in calibrated output were minimal as expected.” Finally, in question thirteen, list the location and name of files. Make sure that the files are readable by anybody so we can get a copy of them . List all the reference files that are being delivered. For example,

```
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp03_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp04_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp05_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp06_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp07_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp08_hrc_rdrk.fits
-rw-r--r-- 2 acsref 10532160 Mar 24 23:13 grp09_hrc_rdrk.fits
```

Mailing the form to the INS/CDBS Group concludes the reference file creator/tester work. Once the files are in the system, the person delivering the file will be notified.

References

- C. Cox & C. Tullos CDBS 1998, "*Delivering Calibration Reference Files*".
- R. Diaz-Miller, STIS TIR 2003-01, "*Assessment and Delivery of Spectrographs Calibration Reference Files*".
- B. Simon, CDBS 1996, "*certify*".