SPACE TELESCOPE SCIENCE INSTITUTE
Operated for NASA by AURA

Technical Instrument Report CDBS 2009-01

Assessment and Delivery of Reference Files

R. I. Diaz, M. Cracraft
rdiaz@stsci.edu, cracraft@stsci.edu

November 10, 2009

Abstract

This TIR defines the standard procedures for the delivery of calibration pipeline reference and SYNPHOT/pysynphot data files. It clearly establishes the instrument and OTA teams' responsibilities and provides guidelines for testing and verification of reference files. This is a revision to TIR 2008-01. In this revision we clarify several of the steps and add new procedures for the format of the DESCRIP keyword. We also provide more examples in the different sections of the document and detail on how to set your environment to run the `fitsverify` *script. We state the policies and procedures for the changes and delivery of the TMT table. We also update information to account for changes in the testing procedures for SYNPHOT/pysynphot files and the detailed information we need from the teams.*

Introduction

The generation of calibration reference files and SYNPHOT/*pysynphot* throughput tables 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) Team 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, while 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 the selection of the correct reference files for a specific set of observations. The selection is based on data file keyword values outlined in ICD-47, located at http://www.stsci.edu/instruments/observatory/cdbs/documents/ICD-47_RevF.pdf or http://www.ess.stsci.edu/projects/distribution/ICD47/ICD-47_RevF.pdf. In the case of the SYNPHOT/*pysynphot* throughput tables, these are used indirectly by the OPUS pipeline via SYNPHOT. Although these files are selected via a Master Component table (TMC table), the selection of the correct TMC table is done much in the same way as any other reference file.

There is another set of SYNPHOT/*pysynphot* files that are not ingested into the CDBS database, but they are considered part of the CDBS system because they are used by SYNPHOT/*pysynphot*. These files are the stellar atlases, stellar libraries, and HST calibration spectra. Although these files do not go to the database, they should also be validated and tested. Currently the files are tested for compliance with CDBS requirements by the CDBS Team, however, since these files affect all the instruments, validation should be done by SSB and all the instrument teams.

In general, 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 Team. If atlases and libraries or HST calibration spectra are being delivered, the INS/CDBS group will work with SSB and the instrument teams to create them and assess their accuracy.

Checking and quality assessment of reference files is required for the following reasons:

- to ensure correct keyword content and syntax
- to ensure correct file structure and format
- to ensure that the files will work properly with the pipeline calibration software
- to ensure that the files are correctly selected by SYNPHOT/*pysynphot*
- to verify scientific validity
- to ensure proper documentation of the reference files

Accordingly, we define in this document the standard procedures for checking and assessing the quality of the 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 while testing the reference files or SYNPHOT/*pysynphot* data files, they must be resolved before proceeding to the next step. The last step, of course, is the delivery of the files to the CDBS team; which further test and assess the compliance of the files with CDBS requirements. Problems with the CDBS tasks should be referred to the INS/CDBS Team (cdbs_team@stsci.edu). The detailed steps and procedures for ingesting reference files into the databases are described in TIR CDBS 2009-02.

Summary: Preparation and Assessment of Reference Files

1. Create or update all the reference files' headers properly.
    - all this information should be in the extension [0] of the reference file
    - make sure the DESCRIP keyword is 67 characters long
2. In the case of pipeline reference tables, update the Comment and Pedigree columns if applicable.

3. Rename files so they contain no capital letters and in the case of SYNPHOT/*pysynphot*, change to the new version number.

4. If you are delivering thermal component SYNPHOT/*pysynphot* files, you have to also create the TMT table.

5. Run fitsverify on the FITS files.

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

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

8. In the case of table pipeline reference files, make sure that all the necessary rows are present.

9. Test the files using CALXXX or SYNPHOT/*pysynphot* and if possible with the ETCs.

    - Use the compare_table IDL procedure for table pipeline reference files

10. In the case of WFPC2 Synphot files create the ASCII version of the files

11. In the case of WFPC2 GEIS files, create the Linux version of the files

12. Fill out the delivery template and notify the INS/CDBS Team that the files are ready for delivery.

Detailed description of the preparation steps

1. Update all the reference files' headers properly

The reference files should be FITS compliant reference files with a primary header, or extension [0], that will have all the relevant information that describes the file being delivered. (All reference files, tables or images, should have a primary header.) Most of the relevant information used by the CDBS system to correctly select the reference file for the calibration of HST data, is contained in a set of header keywords in the primary header. If the reference file being delivered will be replacing an old file already in the system, special care should be exercised when updating these keywords, so the old files are correctly replaced and used by the affected data. The relevant header keywords are:

`USEAFTER` date
`PEDIGREE` of the file
`DESCRIP`tion
`COMMENT`
`HISTORY` lines.

Header keywords that are instrument specific, like `DETECTOR` or `FILENAME`, should be filled in compliance with the formating outlined in the 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`. For SYNPHOT/*pysynphot* data files (also known as throughput tables) three extra keywords have to be present:

```
INSTRUME
COMPNAME
DBTABLE
```

The `USEAFTER` and `PEDIGREE` (and `INSTRUME` and `DBTABLE` for SYNPHOT/*pysynphot* data files) keywords will most likely already exist in the header. `DESCRIP` and `COMMENT` (and `COMPNAME` for SYNPHOT/*pysynphot* 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/*pysynphot* software. In order to avoid confusion, each of these cases is discussed independently. Also, if the reference file's 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.

1.1 USEAFTER (for Pipeline Reference Files)

This is the most crucial of all the header keywords in a reference file that will be used by the calibration pipeline. 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 time. Furthermore, although most reference files are "back dated" 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 to make sure that they will correctly 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 must be used for the new reference file.

1.2 USEAFTER (for SYNPHOT/*pysynphot* Data Files)

The format for this keyword is "`Month dd yyyy hh:mm:ss`", where the time entry is optional. The `USEAFTER` keyword in SYNPHOT/*pysynphot* data files (also known as throughput tables) is used in a different way than the `USEAFTER` date keyword in the pipeline reference files. The

SYNPHOT/*pysynphot* software itself does not care about this keyword, but it is used by the CDBS database to determine when the files were created. It is also used by one of the CDBS tools, the `MKCOMPTAB` task, that creates the Master Component Table (MC table), or "lookup table". This task is used by the INS/CDBS Team in the delivery process to recreate the MC table. (Note that this task is not for the purpose of testing. In these cases the MC table should be edited manually; for example, with the task `TEDIT`.) When the CDBS Team uses the `MKCMPTAB`, the task looks into the CDBS database and identifies the SYNPHOT/*pysynphot* files with the most recent `USEAFTER` dates. The most recent files are then used to populate the MC table for each of the SYNPHOT/*pysynphot* components.

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 list all of them in the TMC table and therefore there will be repeated entries for the same SYNPHOT/*pysynphot* component. In this case, the order in which those files appear is important. SYNPHOT will use only the last of those data files listed in the MC table while *pysynphot* will use the first file listed. Because of this duality, duplicated entries should be avoided to prevent any problems.

So, in order to avoid any possible confusion, the `USEAFTER` date of SYNPHOT/*pysynphot* throughout tables should be set to the date the file was created or the date of the delivery. In no case should this date be left to that of the old SYNPHOT/*pysynphot* file.

1.3 PEDIGREE

The header keyword `PEDIGREE` describes the type of data used to create the reference file. The valid values for this keyword are

`INFLIGHT` dd/mm/yyyy dd/mm/yyyy
`GROUND`
or
`DUMMY`.

If `PEDIGREE = INFLIGHT`, the whole range of dates of the data used to create the file should be listed. For example, "`INFLIGHT 21/01/2000 26/01/2000`"; where the dates have format *dd/mm/yyyy* (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. Also, when working with tables, if there is a pedigree column, update this value for the rows that were changed/affected. More on this in next section.

1.4 DESCRIP

The `DESCRIP` (description) keyword should be a concise sentence; one that makes it easy for users to decipher the type of reference file or the changes being made. This keyword should be a one-line explanation of why the file is being delivered. For example: `DESCRIP = "Reference`

superdark created from proposal 7276", is aceptable as it provides a simple description of the file. However better descriptions would be:

```
DESCRIP = "In-flight low-order flat based on observations of GD 71"
DESCRIP = "Revised sensitivities for first order M modes"
or
DESCRIP = "New blaze function corrections and sensitivities for eschelle modes".
```

More complete information about the changes and how the file was created should be provided in the HISTORY lines.

Why it is important to sumarize the changes made to the files in the DESCRIP keyword? This is important because the content of the DESCRIP header keyword is the first thing to appear in the database parameter that contains the information about the file. That is, it is the first line in the field COMMENT in the CDBS database. The content of the HISTORY section is then concatenated to it. For example, the DESCRIP and HISTORY of reference file t7f19550i_pfl.fits is:

```
DESCIPT 'FQ889N/FQ937N quad pflat created from cold TV3 data ---------------'
HISTORY CCD parameters table:
HISTORY reference table iref$t291659mi_ccd.fits
HISTORY GROUND
HISTORY UVIS-1 CCD characteristics from TV3 data
HISTORY From TV3 data
HISTORY Uncertainty array initialized.
HISTORY DQICORR complete ...
HISTORY WFC3 normal overscan CCD data compatible
HISTORY Dark created from WFC3/UVIS TV3 frames for -82C, MEB2.
HISTORY CR rejection params found in COSMIC RAY REJECTION ALGORITHM PARAMETERS
HISTORY section of header.
HISTORY New quad pflats based on TV3 data.  These pflats were generated with
HISTORY calwf3 (v 1.4.1 as opposed to v 1.0) and using a new *crr.fits file
HISTORY where the sky subtraction value is set to none.  Additional changes
HISTORY include the newest bad pixel table, ccd table, superbias, and
HISTORY superdarks.  CCDAMP keyword value set to ANY and APERTURE keyword set
HISTORY to FULLFRAME_4AMP to accommodate update to CDBS expansion rules.
HISTORY Created with mixed MEB1 and MEB2 flat fields
```

will translate into the entry of the CDBS database Comment keyword:

```
"FQ889N/FQ937N quad pflat created from cold TV3 data --------------- CCD
parameters table:  reference table iref$t291659mi_ccd.fits GROUND UVIS-1 CCD
characteristics from TV3 data From TV3 data Uncertainty array initialized.
DQICORR complete ...  WFC3 normal overscan CCD data compatible Dark created
from WFC3/UVIS TV3 frames for -82C, MEB2.  CR rejection params found in COSMIC
RAY REJECTION ALGORITHM PARAMETERS section of header.  New quad pflats based on
TV3 data.  These pflats were generated with calwf3 (v 1.4.1 as opposed to v 1.0)
and using a new *crr.fits file where the sky subtraction value is set to none.
```

Additional changes include the newest bad pixel table, ccd table, superbias, and superdarks. CCDAMP keyword value set to ANY and APERTURE keyword set to FULLFRAME_4AMP to accommodate update to CDBS expansion rules. Created with mixed MEB1 and MEB2 flat fields"

This information in turn will be then used to fill the Comment section of the CDBS dynamic reference file tables (http://www.stsci.edu/hst/observatory/cdbs/). In this case only the first 67 characters of this field are displayed. To make this field more informative to users, we will now require the instrument teams to provide the reason for the delivery in the DESCRIP keyword. And for clarity, we will also request that deliverers add dashes (or any symbol other than blank spaces or tabs) to this keyword. If the content is less than 67 characters, dashes should be added such that the text plus the dashes reaches this length. When checking the headers, the CDBS team will make sure that the DESCRIP keyword is 67 characters long. If the keyword is not in this format, the files will be returned to the deliverer to be corrected.

In order to facilitate padding the length of this keyword, you could use the *python* script length_descrip.py. This script will check if there are 67 characters in this header keyword and if not it will pad it with dashes in order to make it 67 characters in length. In order to run this script just type:

```
python length_descrip.py
```

In the command line. If your DESCRIP keyword is 67 characters in length, the program will state that all is O.K. and no error message will be printed. If the keyword has more than 67 characters the program will display an error message stating this and the file needs to be modified to contain only 67 characters. A copy of this script is in SMALLS in the directory /calib/cdbs_delivery/useful_scripts/, in the centralized storage in directory /grp/hst/cdbs/tools/useful_scripts/, or in Appendix A of this document.

1.5 COMMENT

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

Note, however, that some FITS reference files have a default COMMENT section that refers to the FITS file format and which cannot be modified or erased. The FITS default COMMENT section is different than the COMMENT header keyword referred to here. The way to distinguish between these two is by their format. In the case of the required CDBS COMMENT line, the word COMMENT is followed by an "=", as in the example above and should list the people who created the file. If this does not exist in the current FITS file, it can be added in one of two ways:

1. Using IRAF, you can first delete the default COMMENT lines that appear in the file and then add the new one. The commands needed to do this are:

   ```
   cl> thedit file.fits[0] comment delete+
   cl> hedit file.fits[0] comment "= 'comment string'" add+
   ```

Note that the "=" should be added at the beginning for it to comply with CDBS requirements.

2. The other way to add the header keyword is by using Pyraf as follows.

```
import pyfits
hdulist=pyfits.open(myfile.fits,mode=update)
hdulist[0].header.add_comment(= comment string,before=origin)
hdulist.flush()
```

This will add a header keyword `COMMENT` leaving intact the FITS comment section already in the file.

1.6 DBTABLE (for SYNPHOT/*pysynphot* data files only)

The `DBTABLE` keyword should be set to "CRTHROUGHPUT" for all SYNPHOT/*pysynphot* throughput files.

1.7 COMPNAME (for SYNPHOT/*pysynphot* data files only)

The `COMPNAME` keyword should have the SYNPHOT/*pysynphot* name of the component to which the delivered reference file applies. That is, the `COMPNAME` keyword in the HEADER of the file has to match the TMG and TMC `COMPONENT` name. For example, if the TMG and TMC files have COMPNAME = ir_f126n, the COMPNAME header keyword of the file to be used should be "ir_f126n". Check previously delivered files to make sure that you are using the right value. For some files, this is the first part of the name of the file you are delivering. For example, the data file "stis_a2d8_001_syn.fits", has `COMPNAME stis_a2d8`. The important thing is that there is agreement between the `COMPNAME` header keyword and the `COMPNAME` column value; i.e. they should be identical.

1.8 INSTRUME (for SYNPHOT/*pysynphot* data files only)

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

1.9 HISTORY

The `HISTORY` lines can be added to the FITS file header in many different ways, and only some of them will be mentioned here. In the case of WFPC2 GEIS files, the header is an ASCII file and it can be edited using any text editor.

In the case of FITS files, one way to add the history section is via the IRAF task called

stfhistory in the stsdas.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; e.g. 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, 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 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 lengthly 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 `eheader`. The IRAF `stsdas.toolbox.eheader` task will allow you to edit the primary image header using an interactive text editor.

1.1 Special considerations for WFPC2 GEIS files

In the case of GEIS files (e.g., WFPC2 reference files), make sure that the headers are FITS compliant, even if those are ASCII files only. For this, replace tabs with spaces and double quotations with single quotations; this is because double quotations and tabs are not FITS compliant. In the case of single quotations, make sure that enough empty spaces are left between the single quotes by matching the format of the rest of the rows in the header file. Following

these guidelines will make sure that all the header keywords are FITS compliant when the files are converted to "waiver" FITS files.

2. In the case of tables, update the Comment and Pedigree columns

For some of the pipelines, information regarding the reference files used for the calibration of the data is provided in the trailer files. The usual information provided in the trailer files are the name of the reference file and information on their pedigree and comment or description. In the case of image reference files, and some of the table reference files, the comment and pedigree that appear in the trailer file are extracted from the header keywords of the file. In the case of table reference files that have a "pedigree" and/or "comment" column, the pedigree and comment section of the trailer file are extracted from the particular row used. Therefore, it is recommended to pay particular attention to the content of these two columns when updating these table reference files. It is important to make sure that the information in these columns is accurate (in the case of "pedigree") and provides a clear description of the data contained in the row and that is relevant for the user to know when reviewing the trailer files. An example for one of the `*_pht.fits` reference tables for STIS would be:

Label	OPT_ELEM	CENWAVE	NELEM	WAVELENGTH	THROUGHPUT	PEDIGREE	DESCRIP
1	G750M	5734	500	5443.	0.081832	INFLIGHT 07/02/1997 14/04/1998	Keyes April, 1998 calibration
2	G750M	6252	500	5959.	0.118651	INFLIGHT 27/02/1997 14/04/1998	Keyes April, 1998 calibration
3	G750M	6768	500	6475.	0.138889	INFLIGHT 27/07/2009 14/08/2009	Updated with SMOV data.
4	G750M	7283	500	6988.	0.133596	INFLIGHT 27/02/1997 14/04/1998	Keyes April, 1998 calibration
5	G750M	7795	500	7499.	0.133596	INFLIGHT 27/02/1997 14/04/1998	Keyes April, 1998 calibration

3. Rename files so they contain no capital letters or version number.

CDBS cannot process files with names containing capital letters. Such files should be renamed to contain only lower case letters. Also, in the case of SYNPHOT/*pysynphot* data files, rename the files by incrementing the value of of the Synphot files by at least one. For example, replace the file name `stis_g750l_007_syn.fits` with `stis_g750l_008_syn.fits`.

4. Verify that the files are in standard FITS format (not applicable to GEIS files)

Mac users should be able to mount the centralized storage area in order to use the CDBS test scripts. In order to correctly define the paths and use the scripts, each user should follow the directions on the 'Scripts and Environment Setup for Mac Users' page created by Paul Goodfrooij. `(http://www.stsci.edu/wiki/CSSC/ScriptsAndEnvironmentSetupForMacUsers)`.

If you are working in the Science (Solaris) cluster you can use the CDBS tools described in the following sections in the UNIX shell line. For this, the path to the CDBS tools

(`/grp/hst/cdbs/tools/bin/`) has to be defined in your private directory `PATH` (in the .setenv file), otherwise the whole path to the script has to be used.

FITS reference files that are not in standard FITS format cannot be ingested into the databases. To verify that these are compliant, run the `fitsverify` tool on the files. For this use the command:

```
fitsverify filename.fits
```

in the `smalls.stsci.edu` domain, or

```
farris_fitsverify filename.fits
```

in the science cluster. Wild-cards may be used for filenames (filename.fits in this example can be replaced by *.fits). The reason for using these two versions in these two different locations is because the `fitsverify` script was updated recently to fitsverify version 4.13 (which is compliant with the GSFC Heasarc version, distributed by Bill Pence). Because of changes in formatting in the new script, it might not work for some of the old format FITS files. If possible, correct the FITS files to be compliant with the new `fitsverify` version; but, you should keep in mind that the CDBS software only requires that these are compliant with the Allen Farris `fitsverify` version. Note that in the `smalls.stsci.edu` domain the `fitsverify` and `farris_fitsverify` versions are the same.

On the Solaris science cluster, some users have problems running the `farris_fitsverify` script. The error we have seen looks like the following:

```
ld.so.1: farris_fitsverify: fatal: libstdc++.so.5: open failed: No such
file or directory Killed
```

If this is the case for you, the easiest solution is to enter the following line:

```
Package: GCC322
```

into your file .envrc in your home directory.

An alternative solution for csh/tcsh users, especially for those of you who do not use the "checkenv" program in your shell setup files (this is not common), is to enter the following command in the file .setenv in your home directory:

```
setenv LD_LIBRARY_PATH /usr/local/gcc-3.2.2/lib:$LD_LIBRARY_PATH
```

for example near the top of .setenv, where other PATHs are defined. In either case, you will then be able to run `farris_fitsverify` when you start a new shell (log in again, or open a new xterm).

The output of these two versions will differ considerably. A sample output for the `fitsverify` (v4.13) for the FITS binary table "p9r19206o_tds.fits" looks like this:

```
fitsverify 4.13 (CFITSIO V2.510)
--------------------------------
```

File: p9r19206o_tds.fits

2 Header-Data Units in this file.

========== HDU 1: Primary Array ==========

49 header keywords

Null data array; NAXIS = 0

========== HDU 2: BINARY Table ==========

48 header keywords

TDS(1) (10 columns x 6 rows)

Col#	Name (Units)	Format
1	OPT_ELEM	12A
2	WAVELENGTH (angstrom	60D
3	TIME (MJD)	12D
4	SLOPE (percent per y	720D
5	NWL	1I
6	NT	1I
7	REFTEMP (K)	1D
8	TEMPSENS (percent /K	60D
9	PEDIGREE	67A
10	DESCRIP	67A

++++++++++ Error Summary ++++++++++

HDU#	Name (version)	Type	Warnings	Errors
1		Primary Array	0	0
2	TDS (1)	Binary Table	0	0

*** Verification found 0 warning(s) and 0 error(s). ****

On the other hand, output from the `farris_fitsverify` 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 (18/12/00 instead of the correct format Dec. 18, 2000)
- 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.

5. If delivering `*th.fits` SYNPHOT/*pysynphot* files, you have to also create the TMT table.

The TMT table is the lookup table for the HST component thermal characteristics. It is the equivalent of the TMC table (The Component Lookup table) for the throughput tables and bandpass tables. Therefore, it should be updated whenever a new SYNPHOT/pysyphot thermal emissivity table (`*_th.fits`) is delivered to CDBS. There is no automatic script to make this update so it is the responsibility of the instrument teams to modify this table and include it with the delivery of the `*_th.fits` files.

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

The CDBS `certify` tool performs additional checking on the syntax and keyword values in the FITS files or the GEIS header files (WFPC2 uses GEIS files), 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 the following command in a UNIX shell.

`certify filename`

Wildcards may be used for filenames, e.g., **.fits* or *.*h. 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. More detailed documentation on the `certify` task is available in postscript format on the CDBS web page (http://www.stsci.edu/hst/observatory/cdbs/documents/). A complete list of the standard instrument-dependent header keywords can be found in ICD-47.

The required keywords are specified to `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 STScI science cluster and the `smalls.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	

A sample of the output of `certify` for a file with no errors will be:

```
== Checking mama2_PFL.fits ==
```

while the output 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, and that the list of required and valid values for the header keywords are present (see ICD-47 for a complete list). If this check does not reveal the source of the error, contact the INS/CDBS Team.

7. 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/*pysynphot* 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 the data 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 strongly recommended for the testing 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 and error extension consistent with the data? Keep careful notes on how you determined that the new reference file was good. You will need this information when you deliver the reference files to the CDBS team. 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.

8. For table PIPELINE reference files, check all rows are present and indicate which changed.

This step is necessary to ensure that no rows were accidentally deleted and that all of the new rows were added. This will also help in gathering information about the rows that have changed

and that should be listed in point 11a of the delivery form. Knowledge of the rows that changed, and their level of change, is now required to facilitate the identification of the data that should be re-processed after a delivery. This is in particular important for the CADS, WCF and the STScI static archive data; which is not reprocessed on the fly.

A script called `compare_table.pro` was developed for this testing. This script can be found in the Science Cluster in `/grp/hst/cdbs/tools/useful_scripts/` or in *SMALLS* in the directory */calib/cdbs_delivery/usefulscripts/*. More information on this procedure is located at *http://www.stsci.edu/hst/observatory/cdbs/deliveries/compare_table_pro*. This script compares two FITS reference file tables, looking for missing elements by checking the differences in each row between the new and old reference files. The calling sequence for the procedure is shown below:

```
idl>compare_table,'my_new_file.fits','old_reference_file.fits',$
COLUMNS=['ccdchip','filter1','filter2'],SAVEFILE=1
```

where 'my_new_file.fits' is the name of the new reference file, 'old_reference_file.fits' is the old reference file, 'ccdchip, filter1, and filter2' are the columns to be compared, and the optional SAVEFILE parameter. The comparison elements are limited to those given in the COLUMNS parameter, which must be specified for the program to run. The parameter COLUMNS is an n-element array of strings with the names of the columns to compare within the table. When a combination is missing in either the old or new file, the procedure will flag it in the standard output. If the parameter SAVEFILE is set to a value other than 0, the output of this procedure will be saved in a file called "compare_table.out".

Any changes should be examined to be sure they were the intended changes. These differences should be listed in point 11a of the delivery form when the files are delivered to CDBS.

9. Test the reference files using CALXXX or SYNPHOT/*pysynphot* and ETC.

CALXXX must be used to calibrate actual data with the reference files being prepared for delivery. Make sure that the CALXXX version used is the one that is being used by the OPUS pipeline. A list of the versions that are being used by different OPUS builds can be found at `/grp/hst/cdbs/versions`. 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 are actually 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/*pysynphot* data files, on the other hand, have to be tested using the SYNPHOT, *pysynphot*, and ETC software (three independent tests using different tools). The ETC software relies heavily on the *pysynphot* package to predict count rates and S/N values via calls to *pysynphot* 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/*pysynphot* data files should also be tested against the ETC software. Unfortunately,

the testing environment that was used for the ETC tests is not currently working and testing against the ETCs software should not be done now. Once the test area is working again, or a new system is created, this document will be updated and the instrument teams will be notified.

9.1 Testing the file in SYNPHOT/*pysynphot*

The testing under SYNPHOT is accomplished using the `stsdas.hst_calib.synphot` package in IRAF, while the test of *pysynphot* is done using *pyraf* in *python*. Here, it is also assumed that the files are being tested in the STScI Science cluster or in a system that mounts the centralized storage area /grp. If working on a different platform, the paths to the directories mentioned below will change.

1. The first step of this process is to create a dummy "Master Component Table" (TMC table) that points to the data files you are delivering. For this, make a copy of the most current TMC table in a working directory. All the TMC tables are located in the directory `/grp/hst/cdbs/mtab/` in the centralized storage area. Although there are several TMC tables in this directory, the most up to date is listed last in alphanumeric order. Choose the last one listed when sorted alphabetically.

2. Now edit the row(s) for the COMPONENT(s) associated with the data file(s) you are delivering. Make sure that there is one row in this table for the same COMPONENT name. If you find repeated entries, delete rows to leave only one; usually the one with the greatest number of the set. Change the `FILENAME` column of this row to point to the SYNPHOT/*pysynphot* data file being delivered. You can edit this file via the IRAF task `tedit`. Note that there is a required 68 characters limit for this column. In order to prevent a problem with long path names, as well as for the use of this file for the testing with the ETC software, we strongly recommend defining a logical IRAF variable for this directory. In the ETC test environment this file has to be called "comptest", so we recommend using this name here as well. In this case the `FILENAME` column value will have the following format:

   ```
   sy> comptest$rootname_001_syn.fits
   ```

   With SYNPHOT

   (a) In order to test the files with SYNPHOT, the IRAF logical variable "comptest" also has to be defined. This can be accomplished by typing in the IRAF command line:

   ```
   sy> set comptest = ‘‘/data/mypath/myworkingdir’’
   ```

   (b) Now modify the parameter "cmptbl" in the `stsdas.hst_calib.synphot.refdata` task with the name of the dummy TMC table using the same IRAF logical variable name to point to the working directory.

   (c) If you are delivering a "Master Graph Table" (TMG table) together with your new SYNPHOT/*pysynphot* data file(s), you also have to modify the parameter "grtbl" of the `refdata` task. This is done in the same way as for the TMC table.

(d) Now run the SYNPHOT/*pysynphot* task that will use this data file. Note that if you're also delivering the "HST Thermal Component Table (TMT table), this cannot be tested at this point.

(e) Make sure that you used the correct SYNPHOT/*pysynphot* data files by running the task `stsdas.hst_calib.synphot.showfiles`.

(f) If you are delivering a new TMG file, you will also have to run the task `synphot.grafcheck` to validate the structure of the graph table (e.g. `grafcheck qc80137jm_tmg.fits`) For more information on the SYNPHOT/*pysynphot* package and tasks, refer to the *Synphot User's Guide*. (http://www.stsci.edu/resources/software_hardware/stsdas/synphot/SynphotManual.pdf)

With pysynphot

Here we provide instructions on how to use the `plband` in *pysynphot* as well as the *pysynphot* equivalent to `showfiles` and `countrate` in SYNPHOT.

(a) Set the path for the CDBS directory by typing in your command line

```
mycomputer>setenv PYSYN_CDBS /grp/hst/cdbs/
mycomputer>irafdev
```

(b) Start a *python* session by running:

```
mycomputer>python
```

(c) Import *pysynphot* and *matplotlib*

```
import pysynphot as S
import matplotlib.pyplot as P
```

Note that you do not have to use S or P as you can set the variables to anything you want.

(d) Check which TMC, TMG, and TMT file you are using. For this type in *python*:

```
S.showref()
```

The output to this command will be; e.g.:

```
thermtable:  /grp/hst/cdbs/mtab/tae17277m_tmt.fits
comptable:  /grp/hst/cdbs/mtab/tad1851am_tmc.fits
graphtable:  /grp/hst/cdbs/mtab/t2605492m_tmg.fits
area:  45238.93416
```

To change the `comptable`, `graphtable`, or `thermtable` parameters use the following command:

```
S.setref(comptable='/mydir/test_tmc.fits')
S.setref(graphtable='/mydir/test_tmg.fits')
S.setref(thermtable='/mydir/test_tmt.fits')
```

(e) Now set the bandpass to be analyzed using the following *pysynphot* command:

```
bp = S.ObsBandpass('cos,fuv,g130m,c1291')
```

in this case we want to set the bandpass for COS FUV with grating G130M and central wavelength 1291 Å. In order to see what files will be used, you have to use the *pysynphot* command `showfiles`

```
bp.showfiles()
```

the output of this command is; for example:

```
/grp/hst/cdbs/comp/ota/hst_ota_007_syn.fits
/grp/hst/cdbs/comp/cos/cos_psa_002_syn.fits
/grp/hst/cdbs/comp/cos/cos_fuv_correction_002_syn.fits
/grp/hst/cdbs/comp/cos/cos_g130m_003_syn.fits
/grp/hst/cdbs/comp/cos/cos_mcp_g130mc1291_005_syn.fits
```

(f) Now use the *matplotlib* `plot` task to make sure that *pysynphot* is correctly picking up the file. For this run the command:

```
P.plot(bp.wave,bp.throughput)
P.xlim(x1,x2)
```

were `P.xlim()` sets the wavelength range.

(g) You can also calculate the total count rate for this mode by using the *pysynphot* command `countrate`. The first step is to set the spectrum to use. This can be done with the *pysynphot* command `FileSpectrum`. For example:

```
sp = S.FileSpectrum('$PYSYN_CDBS/calspec/gd71_mod_005.fits')
```

Now set the observation mode to be used in the calculations with the following *pysynphot* command:

```
obs = S.Observation(sp,bp)
```

Finally run the `countrate` command:

```
obs.countrate()
```

The output of this command is a single floating point value with the total count rate for the selected mode.

We recommend running the same *pysynphot* commands using the default TMC, TMG, and TMT files and comparing these with the new values. For more information on how to run other *psynphot* tasks refer to the *"Introduction to Pysynphot for Synphot Users"* guide in
http://mediawiki.stsci.edu/mediawiki/index.php/Introduction_to_Pysynphot_for_Synphot_Users.

3. Once the real differences are understood, the test is completed and the new SYNPHOT/*pysynphot* data file(s) can be delivered. Send, along with the delivery form, the location of the `./xCheck` run, documentation on the differences found, and appropriate explanation on how these differences relate to the new delivered SYNPHOT/*pysynphot* data file(s). This information is needed by the INS/CDBS Team to assess compliance with CDBS guidelines.

4. In the case of WFPC2 Synphot throughput tables, the WFPC2 Team also has to deliver the ASCII version of these files. The CDBS Team has an IDL procedure that can be used for this purpose and given in Appendix B of this document.

   This IDL procedure, `convert.pro`, written by M. Wolfe, converts FITS files with two columns: Wavelength and Flux, to ASCII files. The input for this procedure is a file called *input_list*. This file can contain one or more files to convert. The output will be written to a file with the same root name as the FITS file but with extension *.ascii*.

   If the file to be converted is a Synphot Throughput table, the procedure file has to be edited to read column `data[j].throughput` rather than `data[j].flux`. Put the `input_list` file and the script in the same directory as the files to be converted and run it. Note that this script will not work for files that have more than two columns.

10. 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 Team (cdbs@stsci.edu). This template, besides helping the INS/CDBS Team to correctly deliver 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 can also be found in the INS/CDBS web page under Delivery Notification Template (http://www.stsci.edu/hst/observatory/cdbs/deliveries/delivery_form.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/pysynphot?  (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.):
11a- If files are tables with information about different observing modes, please
indicate exactly which rows have changed.
12- Description of how the files were ‘‘tested’’ for correctness:
13- Additional considerations:
14- Reason for Delivery:
15- 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. The field "Date of delivery" should have the date when you deliver the file to the INS/CDBS Team. 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 are usually 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. We need 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 11. 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. 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 were to correct minor mistakes only and they will not impact or will have negligible impact on the calibrated products, the change level is considered TRIVIAL.

Note that in the case of tables we will need more detailed information on which rows changed. This is particularly important for tables that contain information about several observing modes. For these, we need to know which datasets will be affected by the delivery; so the mirror sites and the HLA, can identify with confidence which particular datasets will need recalibration. For these cases, please provide the list of rows that were changed. You can determine which rows were changed by running the script `'compare_table.pro'` discussed in section 7 above. The script is located in the Science Cluster in the STScI IDL area. If you do not have that set in your IDL PATH, you can also find a copy of the procedure in the centralized storage in the directory `/grp/hst/cdbs/tools/useful_scripts/`. This list of changed rows should be provided in line '11a' in the delivery form. Note, however, that in the case of SYNPHOT/*pysynphot* tables, we do not need to know the particular rows that changed.

We also need information on how the files were "tested" for correctness. In question 12, 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."

In question 13, describe any special needs associated with the delivery. Is the installation meant to be coincident with a specific OPUS build or are there other timing requirements or urgency? Note that the INS/CDBS Team makes every effort to deliver your files to OPUS within 24 hours of receiving the delivery form, and will contact you if this schedule cannot be met.

In question 14, please provide with detailed information regarding the reason for which the files are being delivered, i.e. "Update gains for instrument combination X and Y" or "WFPC2 darks to be used for data taken after Dec. 1, 2008". This information will be used later to notify users about the new reference file and the reason for which it was delivered (see next section). We leave it to the instrument teams to provide us with the information that they think will be adequate for this message.

Finally, in question 15, give the directory path where the files are located. Also, provide a complete list of the files delivered. 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 Team concludes the reference file creator/tester work. Once the files are in the system, the person delivering the file will be notified via an e-mail.

11 Information about post delivery steps by the CDBS Team

As one of the final steps in the delivery process the CDBS Team will send a message to users subscribed to the Reference Files Update mailing list about the availability of new reference files. As the HST user will most likely be interested in the instrument they are working, we have created mailing lists for each of the HST instrument teams. The mailing lists are part of the Majordomo system at STScI and have names xxx_reffiles_upd, where xxx is the instrument team name or an abreviation: acs, stis, nic, cos, wf2 (for WFPC2), or wfc3. The message is sent shortly after the files have been ingested into the OPUS pipeline and Archive systems and it provides a link to the CDBS Dynamic Tables for the particular type of reference file.

The messages have a template format for all the instrument teams. As stated before, there

is some information that is added depending on the type of file being delivered or the reason for the delivery. It is important that the instrument teams provide all the necessary information that will help us to give a complete and accurate description about the new reference file and about how and which data will be affected. The template for this file is in `smalls.stsci.edu` directory /calib/cdbs_delivery/notification_form in the CDBS Delivery Procedures webpage (http://www.stsci.edu/hst/observatory/cdbs/deliveries/Reffile_update_notification), or in Appendix C of this document.

Acknowledgements

We would like to thank Michael Wolfe, Sami-Niemi and Tyler Desjardins for providing a script to check the length of the `DESRIP` keyword, and Paul Goudfrooij and Michael Swam for providing ways to make the CDBS scripts work in the different operating systems.

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*"

Laidler et al, 2005, "*Synphot User's Guide*", Version 5.0 (Baltimore: STScI). B. Simon, CDBS 1996, "*certify*"

ICD-47 (http://www.ess.stsci.edu/projects/distribution/ICD47/ICD-47_RevF.pdf)

Appendix A

python Script `length_descrip.py`:

```
import glob
import pyfits as PF
from pyraf import iraf
from iraf import images,imutil

tmp = glob.glob('*.fits')

for file in tmp:
print 'Now processing %s' % file
des = PF.open(file)
hdr = des[0].header
val = hdr['DESCRIP']
print 'Number of Characters is:  %i' % len(val)
if len(val) < 67:
print 'Error:  Not Enough Characters
n Adding characters so length is 67'
l = len(val)
pad = '-'*(67 - l)
new = val + pad
iraf.unlearn('hedit')
iraf.hedit(file+'[0]','DESCRIP',new,update='yes',show='no',verify='no',delete='no')
elif len(val) > 67:  print 'Error:  Too Many Characters
n Please edit DESCRIP keyword to 67 characters'
elif len(val) == 67:  print 'No Error
n'
```

Appendix B

IDL procedure `convert`:

```
pro convert

; convert fits files to ascii files
; list desired files to convert in a file named 'input'

readcol,'input',name,Format='A'
close,1
for i=0,n_elements(name)-1 do begin

p=strpos(name[i],'.')
print,p
file=name[i]
file=strmid(file,0,p)
a="openw,1,'"+ file +".ascii'"
print,a
ss=execute(a)
data = mrdfits(name[i],1)
tags = tag_names(data)
n = n_elements(data.wavelength)
for j=0L,n-1 do begin
printf,1,data[j].wavelength,data[j].flux,format='(2g14.6)'
endfor
close,1
endfor
end
```

Appendix C

Template to announce to the HST community about the delivery of new reference filess:

```
Subject line:  Announcing the delivery of new XXXX reference files.


Dear HST user,

On mm/dd/yyyy the XXXX team delivered a new (set of) reference file(s) to be
used with XXXX data.

The reference file(s) delivered and reason for delivery are:

Filename:
To be used with data taken between dates:
Reason for delivery:

For more information about the modes these files affect and to assess if you
need to recalibrate your data,
please check the reference file pages for the XXXX team at

ACS Links

ACS Bias Images

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/acs_bias_images.html
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/WFCBiasReferenceTest?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCBiasReferenceTest?no_wrap=true

-------------------------------------------------------------------------------------------

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ACSCosmicRayRejection?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ACSCCDTable?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ACSDistortionCorrection?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/SBCLinearity?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ResidualGeomDistortion?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ACSBadPixelTable?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/MultiDrizzleParam?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/ACSCCDOverscan?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/WFCFlatImage?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/WFCPolarizerFlat?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCCoronagraphFlats
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCotherFlats
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCPolarizerFlat
```

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/SBCFlatImage?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCCoronagraphSpotTable?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCCoronagraphSpotFlat?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/WFCDarkReference?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/HRCDarkReference?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/ACS/SBCDarkReference?no_wrap=true

COS Links to reference files

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSFlatImage?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSgeocorr?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSbadTimeIntTable?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSBRF?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSburstParamTable?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COS1dExtract?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSDataQual?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSdeadtime?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSDispRelation?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COScalLamp?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSpulseHeight?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSphotSensitivity?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSTDS?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/COS/COSwaveCal?no_wrap=true

STIS Links to Reference files

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISBias?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDarks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDarksMAMA?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISPFLflats?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDFLflats?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISLFLflats?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISShutterShading?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISSmallScaleDist?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISBadPix?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISCCDpar?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISLinearity?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISA2D?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISPhotConversion?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISApThroughput?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISCalLamp?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISApOffset?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDistCoeff?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISSpectCoeff?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISAngCorr?no_wrap=true

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDispCoeff?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISSpecTrace?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISExtractParam?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISMAMAoffset?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STIScrreject?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISPhotCorr?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISWaveCalPar?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISXDispScatter?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISEchScatter?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISEchXDispScatter?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISHalo?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISRipple?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISHaloLambda?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISDarkCorr?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/STIS/STISOTAPSF?no_wrap=true

WFPC2 Links to reference files

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2Mask?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2AtoDCorr?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2Bias?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2Darks
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/1997andPriordarks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/1998darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2000darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2001darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2002darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2003darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2004darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2005darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/2006darks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2Flats?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2Shading?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2PixCorr
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2IDC?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFPC2/WFPC2OFF?no_wrap=true

WFC3 Links to Reference Files

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3BiasReference?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3UVISdarks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3IRdarks?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3PFLbin1UVIS?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3PFLbin2UVIS?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3PFLbin3UVIS?no_wrap=true

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3PFLflatsIR?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3DFLflatsUVIS?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3DFLflatsIR?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3LFLflats?no_wrap=true

Other files - Links to other reference files

http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3ShutterShading?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3Linearity?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3A2D?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3BadPix?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3CCDchar?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3Overscan?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3crreject?no_wrap=true
http://www.stsci.edu/hst/observatory/cdbs/SIfileInfo/WFC3/WFC3distcoeff?no_wrap=true