Technical Instrument Report CDBS 2008-02

Delivery of Reference Files to the Data Management Systems

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

June 24, 2008

Abstract
This TIR describes the INS/CDBS Team's responsibilities. It defines the standard procedures for the delivery of calibration pipeline and SYNPHOT reference files to the Data Management Systems (DMS). It provides guidelines for test and validation of reference files by the INS/CDBS Team. This is an update to TIR CDBS 2005-02A. It clarifies the procedures to deliver all types of SYNPHOT files, including Atlases and bandpasses.

Introduction

In order for reference files to be used by the OTFR pipeline, these files have to be copied to OPUS disks. Pointers to these files are based on instrument modes and applicability dates. On the other hand, their ingestion in the Data Archive and Distribution Services (DADS), disk media and database, allows users to retrieve them from the archive. The main function of the CDBS database is to allow selection of the correct reference files based on an instrument's configuration and date. The selection is based on data file keyword values and criteria outlined in ICD-47 (http://www.stsci.edu/instruments/observatory/cdbs/documents/icd-47RevF.pdf or http://www.ess.stsci.edu/projects/distribution/ICD47/ICD-47_RevF.pdf.). More information can be found in the CDBS Documents web page (http://www.stsci.edu/hst/observatory/cdbs/documents).

The person creating the file will be checking and assessing the quality of the reference files. The official delivery of the files to CDBS is handled by the INS/CDBS Team. Here we describe detailed steps and procedures for ingesting reference files into the databases.

Summary: Test and Validation of Reference Files

1. Transfer the reference files to a directory in the CDBS delivery area

2. Check the permissions of the files
3. Make sure that all relevant header keywords and history section are present in the header [0] of the FITS file
4. Verify that the files are in standard FITS format

For calibration spectra or spectral models, skip all the steps below and go to appendix B

5. Run the CDBS `certify` tool on the FITS files or GEIS header files
6. Create the "load" files
7. Populate the "load" files
8. Certify the "load" files
9. Run `check_load`
10. Rename the pipeline reference files to have an unique name identifier
    - In the case of SYNPHOT files, make sure that the throughput or bandpass files have a number that is greater than the last delivered version of the file
11. In the case of SYNPHOT bandpasses, make sure all the instrument teams have tested these files. Obtain sign-off from all of them before delivery.
12. Put the files in a delivery directory and deliver them to the DMS
13. Fill out the template delivery form and e-mail it to the DMS (if OPUS_FLAG = Y)
14. Transfer the files to the centralized storage area
15. In the case of SYNPHOT files, create the TMC table and deliver it together with the TMG and TMT files (if applicable)
    - Transfer the TMC, TMT, and TMG files to the centralized storage area for further testing
    - Send a message to Vicki Laidler and wait for acknowledgment that they passed testing
    - Deliver the file to DMS
    - e-mail the delivery form to DMS
16. Run `cdbs_report`
17. Check the size of the files in the archive
18. Check that the files are correctly used in the archive
19. Send notification to deliverer (when applicable)
20. Fill out delivery information on CDBS WIKI

Detailed description of the preparation steps

The following steps assume that you are using the special account for CDBS deliveries and that you are working in the `smalls.stsci.edu` domain. In this domain, the account settings have the appropriate permissions to access the databases and the different disk locations used for the delivery process. We suggest following these steps in the order they appear, and whenever a problem is found in one of them, try to solve it before proceeding with the next step. A log file should be kept documenting the tests done on the files. As a team convention, save the log with the output of the CDBS and other commands in a file named "delivery.log". This file will be used by the team as a status report for deliveries. In the following sections, all the examples provided will be given assuming that the outputs are re-directed to this log file. Given that some of the steps described here will use IRAF tasks, open an IRAF session too.

1. Transfer the reference files to a directory in the CDBS delivery area

Create a delivery directory. Currently the deliveries are done from the `smalls.stsci.edu` domain. The area assigned to test and validate the reference files prior to delivery is located in the directory /calib/cdbs_delivery/. Here, each instrument team has a particular area assigned. For example, ACS deliveries are in the /calib/cdbs_delivery/ACS/ directory while STIS deliveries are in /calib/cdbs_delivery/STIS/. The delivery directories are named after the date when the files were delivered to the CDBS Team, with format *yyyy_mm_dd*, where *yyyy* is the year, *mm* the month, and *dd* the day. These files will remain here for safekeeping until the files are ingested into the databases. Complete steps 2 to 18 in this directory. To save disk space, gzip `lod` files and erase the `FITS` files after the files have been ingested. A log file should be kept with the output of all the scripts and tasks used to test the files. For this, redirect the outputs to a log file using the append redirection command: ">> &". Note the "&" character, this is used to record all the output flags, including those that are sent to the standard error instead of standard output.

Transfer the FITS files from the deliverer directory to this directory using FTP or copy (if they are already in the `smalls.stsci.edu` domain). When using FTP remember that you are transferring binary files and that the transferring mode has to be binary.

In the case of WFPC2 files (except for the IDC reference file), the delivery will have instead of FITS files, four GEIS files (two files "*d" and two files "*h") and one "*.lod" file per delivered reference file. Transfer all to the smalls working directory. Note also that these files do not have the usual extensions (e.g. "drk"). In this case, the format of the files is **rootname**`.r#x`; where `#` can be a digit between 0 and 6, and `x` will be the letter "d" or "h". In the case of WFPC2 "drk" files, the root name of the files should be unique, i.e. the WFPC2 Team has already renamed them using the `uniqname` script. This is because the "drk" files are generated automatically. Other type of files should be renamed with the `uniqname` script by us.

The WFPC2 team delivers two types of GEIS files, one that can be used withe the Solaris and MacOS systems, and another that can be used by Linux systems. Only those used in Solaris systems should be tested and delivered. The Linux files are only transferred to the appropriate directory in the centralized storage area; see step 13.

If the delivery is of SYNPHOT throughput files and you also receive a "Master Graph Table" (TMG; for the file extension name) and the "HST Thermal Components Master Table" (TMT; for the file extension name), make sure that you put the throughput and the TMG and TMT files in different directories, for example, a subdirectory labeled **tmtables**. This is because the TMG and TMT files have to be delivered together with the "Master Component Table" (TMC; for the file extension name) after all the throughput files are in the system. Perform the following steps for the throughput tables only. The TMG and TMT files will be tested later, together with the TMC file.

2. Check the permissions of the files

Using the command `ls -la`, make sure that all the files in the delivery directory have"user", "group", and "other" read permissions. For example, in the following list:

```
-rw-r--r-- 1 srefpipe 31680 May 11 17:31 p5b1731aj_idc.fits
-rw-r--r-- 1 srefpipe 22068 May 11 17:31 p5b1731aj_idc.lod
```

the string `-rw-r--r--` indicates that the files can be read by anybody. This is necessary for the files to be correctly transfered to the OPUS and test areas.

3. Make sure that relevant header keywords and history section are present.

As mentioned in TIR CDBS 2005-01A, there are four header keywords that should be present and correctly populated in all reference files: `PEDIGREE`, `USEAFTER`, `DESCRIP` and `COMMENT`. For SYNPHOT data files, the header keywords: `INSTRUME`, `COMPNAME`, and `DBTABLE` should also be checked. To do this check, the IRAF tasks `hedit` or `hselect`, or command `more` can be used. Examples of `hedit` and `hselect` tasks are:

```
hedit *.fits[0] pedigree,useafter,descrip,comment .
or
hselect *fits[0] $i,pedigree,useafter,descrip,comment yes
```

An example using `more`:

```
more nameoffile.fits
```

In the `more` case, only one file at a time can be checked. To escape `more` mode, type "q". In the case of WFPC2 GEIS reference files, only the GEIS header files have to be checked. The GEIS header files are ASCII files and have extensions ending in "h". In this step, the two header files per dataset have to be checked to make sure they have the same and complete information.

If any of the relevant keywords are missing from the header of the files, contact the deliverer and request the needed information. If the field COMMENT is missing, it can be filled with the name of the deliverer as the creator of the file; e.g.,

"Reference file created by J. Smith."

This can be done using the IRAF command hedit and selecting the add option:

```
hedit filename_xxx.fits[0] COMMENT ''Reference file created by J. Smith.''  add+
```

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 section (header keyword) referred to here. The way to distinguish between these two is by their format. In the case of the CDBS required COMMENT line, the word COMMENT is followed by an "=", as in the example above and should list the people who created the file. For the cases when the FITS COMMENT line exists, the CDBS COMMENT can not be added with the IRAF task hedit, but in the following two ways.

Using IRAF, you can first delete the default FITS 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, or a comment section line would be added rather than the header keyword you were trying to create. If this command does not work, try removing the = in the "= 'comment string'" and just add the comment you want, but double-check the header file to be sure the comment shows up with an equal sign. 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 last one will add a header keyword COMMENT even if a comment section already existed. Another task that will allow you to manually edit the headers of the files if necessary is the IRAF task eheader. This task opens the header in a "vi" environment for editing.

In the case of the history section, check that it has a section relevant to the current delivery. This can be checked by comparing with the information provided in the delivery form. If this information was not provided request it from the deliverer. Check TIR CDBS 2005-01 for the relevant information needed in the "history" section and how to update it within IRAF.

When checking the HISTORY lines, keep in mind that there is a known bug in the CDBS software which could make the delivery fail. The information of the tracking file, known as a "load" file and accompanying each delivered file, is extracted from some of the header keywords and history lines of the FITS file. Before they go in the "load" file, those are stripped of the word HISTORY, extra spaces, and blank lines. Therefore, if a history line has the word "go" at the beginning of a line, it would be mistaken for the "go" SQL command and the ingest will fail with no clear error. Check all the lines of the history to make sure that none starts with this word.

For WFPC2 GEIS reference files, the "*h" files contain the header information. These are ASCII files and can be checked all at once using your favorite text editor or with grep. For example,

```
grep -n USEAFTER *h
```

```
or
grep -n COMMENT *h
```

Check all the header keywords and HISTORY section this way.

4. Verify that the files are in standard FITS format

Although the person creating the reference files has already verified that these are in standard FITS format, double check them, as files that are not in standard FITS format cannot be ingested into the databases. (Note that GEIS files should not be tested with this command.) For this, run the `fitsverify` script on the files:

```
fitsverify filename >>& delivery.log
```

In the STScI Science Cluster, the `fitsverify` version is different than that in `smalls.stsci.edu` domain and some of the files could fail this test even though they pass the `fitsverify` test in `smalls.stsci.edu`. Therefore, in the Science Cluster, the FITS format verification has to be done with `farris_fitsverify`, which is the same one as the `fitsverify` version in `smalls.stsci.edu`.

```
farris_fitsverify filename >>& delivery.log
```

Wildcards may be used instead of file names, e.g., filename can be *.fits. A sample output from this script 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 [1024,1024]
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 this verification include:

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

If any problems are found at this stage, send a message to the deliverer (Cc: cdbs@stsci.edu) with an explanation of the problem. Notify the deliverer that with this message you are canceling the delivery and that you need to receive a new delivery form when the file(s) has been fixed. If you are able to identify the problem include this information in your e-mail. Remember that it is the responsibility of the deliverer to make sure that the delivered files are FITS format compliant.

5. Run the CDBS `certify` tool on the files.

The CDBS `certify` tool performs further checking on the syntax and keyword values in the reference file, 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. In the particular case of WFPC2 GEIS files, only the GEIS header files (extension *.*h) should be run against `certify`. For SYNPHOT Atlas files or HST Calibration Spectra files (CALSPEC), the `certify` tool is not run, as these are not recognized by the CDBS tools. For all the other cases, any errors in this file should be resolved before proceeding with the next step. Note that most CDBS scripts can also be accessed through IRAF in the stlocal.cdbsutil package; but those are likely an older version than the command line versions, so do not use them. When using the tools in this document on a Solaris machine, be sure that it is a Solaris 10 machine (`smalls` is a Solaris 10), as the tests are likely to fail on an earlier version of Solaris. The `certify` tool is run by typing in the command line:

```
certify filename.fits >>& delivery.log
or
certify filename.*h >>& delivery.log
```

Wildcards may be used for filenames; e.g., *.fits or *.*h for WFPC2 header files. More detailed documentation on the `certify` task is available, in postscript format, in the CDBS web page (http://www.stsci.edu/hst/observatory/cdbs/documents/). The `certify` tool does not check all the keyword syntax and values in the reference file, but only those that are specifically used in CDBS, OPUS, and DADS for selecting and tracking the reference files. A complete list of the instrument-dependent standard header keywords can be found in ICD-47.

These required keywords are accessed by `certify` via CDBS template files; template files end with ".tpn". There is a pair of files for each reference file type. One is for the FITS or GEIS files and one is for the "load" files (*_ld.tpn). These files are located in the CDBS working areas of the Science Cluster and the `smalls.stsci.edu` domain. In the smalls domain, these files are currently in the /store/smalls/cdbs/tools/data/ directory, while in the centralized storage the files are located in the /grp/hst/cdbs/tools/data/ directory. Note that whenever a template file is updated in the smalls domain, it should also be updated in the Science Cluster, otherwise the person delivering the file and working in the Science Cluster will not be using the most up to date version. A more detailed explanation on the procedures to change these files will be given in another CDBS TIR. 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 `certify` output for a file that has a problem is:

```
== 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. A complete list of required and valid values for the header keywords can be found in the template files or in ICD-47. If you identify the cause of the error, contact the person delivering the file to describe the problem and solicit input to fix the file. You could also reject the delivery and request the deliverer to send a new delivery form once the reference file has been fixed. In this case, you will have to re-start the process from step 1.

6. Create the "load" file

In order to correctly ingest the files in CDBS, an ASCII "load" (*.lod) file is created for each reference file. This "load" file contains information from the reference file header, and information from the database about existing reference files. Exceptions to this are deliveries for WFPC2 data composed of GEIS files, SYNPHOT Atlas files, and HST Standard Calibration spectra. The process for these files will be explained at the end of this section.

The information in the "load" file is used in the delivery process to create SQL command scripts that populate the databases with the necessary information for the correct selection of the files. The "load" file will have the same root name as the FITS reference file, but with the extension "lod". The file consists of two sections: the header section and the row section. For image reference files, there is one header section followed by one row section. For table reference files there is one header section followed by one or more row sections, each corresponding to a row, or group of rows, in the reference table. The number of rows for table files is usually determined by the selection criteria for the given reference file; therefore, regardless of the number of rows in the table, some table reference files will have several row sections in the "load" file while others will have only one. To create the "load" file type the following command:

```
mkload filename >>& delivery.log
```

Wildcards may be used for filenames, e.g., filename can be *.fits. In the case of WFPC2 GEIS files, filename is the name of the GEIS header file with extension ".r*h", and will be discussed later. An example of a "load" file for a reference file image:

```
FILE_NAME = 11x1_2001_1120_1125_ref_bia.fits

INSTRUMENT = stis
REFERENCE_FILE_TYPE = bia
USEAFTER_DATE = Nov 20 2001 00:00:00
COMPARISON_FILE = lbq12111o_bia.fits
OPUS_FLAG =
```

```
COMMENT =
ENDHEADER

CHANGE_LEVEL =
PEDIGREE = INFLIGHT
OBSERVATION_BEGIN_DATE = Nov 20 2001
OBSERVATION_END_DATE = Nov 25 2001
BINAXIS1 = 1
BINAXIS2 = 1
CCDAMP = D
CCDGAIN = 1
CCDOFFST = 3
DETECTOR = CCD
COMMENT =
ENDROW
ENDFILE
```

The `mkload` command will use information contained in the FITS file to fill some of the fields of the "load" file. There are a few more things about this file and command that are worth mentioning. The `mkload` command automatically fills the `USEAFTER_DATE` field with the `USEAFTER` header keyword information in the FITS file, while the `COMPARISON_FILE` parameter is obtained from the CDBS database. In the latter case, the information from the header and row level information is used to determine the correct comparison reference file. If no file of the same type is found (e.g. when a new type or new mode is being delivered) this parameter will be filled with the value `(INITIAL)`. This prevents other CDBS commands from trying to compare the fields of the new reference file with those of an old file. Note also that in the case when the reference file has a `PEDIGREE` value of `INFLIGHT`, the `mkload` task will populate the `OBSERVATION_BEGIN_DATE` and `OBSERVATION_END_DATE` with the dates given in the FITS file header keyword. More detailed documentation on the `mkload` task is available in postscript format in the CDBS web page (http://www.stsci.edu/instruments/observatory/cdbs/documents/).

In the case of dark and bias WFPC2 files, we receive four GEIS files for each reference file and the "load" file, so we do not have to create it. For other cases, the "load" file should be created using the GEIS header file with extension ending in "r?h" (where ? can be a digit between 0 and 6). In the case of WFPC2 "IDC" reference tables, standard FITS files are delivered and those can be treated as any other FITS reference file mentioned at the beginning of this section.

In the case of SYNPHOT Atlas files (e.g. Kurucz) and HST Calibration Standards spectra, the files are not recognized by the CDBS tools, so the "lod" file cannot be created. These files are not delivered to the CDBS, OPUS, or DADS databases; however, these need to be copied to the corresponding directory in the centralized storage (refer to section 13).

7. Populate the "load" files and check them

The "load" file has several important fields that should be populated. As we mentioned in step 6, some of the fields are automatically populated by `mkload` using the information from the primary and extension headers of the FITS file. Here we will describe those fields for which the content is common to all instruments. These fields are: `OPUS_FLAG` and `COMMENT` in the header section, and `CHANGE_LEVEL`, `PEDIGREE`, `OBSERVATION_BEGIN_DATE`, and `OBSERVATION_END_DATE` fields in the row section. Other fields in the "load" file vary from file to file and therefore will not be mentioned. Note also that an IRAF task, `setlodkeywd`, has been developed to help populate the "load" files and will be explained in detail later in this section.

OPUS_FLAG

Set this to `Y` or `N` to indicate whether the files should be stored in the archive and by OPUS or not. We do expect to deliver reference files that for special reasons, should not be stored in the archive or by OPUS; for example, in the first stages of development, the WFC3 and COS teams requested that the SYNPHOT WFC3 and COS files not be stored in the archive. The `OPUS_FLAG` should be set to `N` for such cases, and the files will not be delivered to the OPUS and DADS databases. This information should be given by the deliverer via the delivery form. **Note** that the `OPUS_FLAG` for the TMG, TMT, and TMC should always be set to `Y`, so the latest version is always stored in the archive.

`COMMENT` *(in the header section)*

The `COMMENT` section in the "load" file is the equivalent to the `HISTORY` section in the data header. The information included here will appear in the StarView forms and on the reference file webpages; therefore, it is recommended to fill this section with information relevant to the delivered file only. This information should be provided by the deliverer or be contained in the `DESCRIPTION` and `HISTORY` section in the FITS reference file.

`CHANGE_LEVEL`

This keyword defines the level of change of the reference file with respect to the last delivered file (given in the field `COMPARISON_FILE`). Note that for table reference files, the changes could affect only a few rows in the file. In this case, only the modified rows should have a value other than `TRIVIAL` (the rows that were not modified should always be `TRIVIAL`); however, in some cases even the change level of the modified rows could also be `TRIVIAL`. In the case of image reference files, there is only one row section. The `CHANGE_LEVEL` should be set to one of three values: `SEVERE`, `MODERATE`, or `TRIVIAL`. The criteria for each are:

SEVERE

i Initial delivery of any file

ii Change that requires existing data to be recalibrated

iii The row-level field for a table has changed by more than 50% compared to the `COMPARISON_FILE`

MODERATE

i Changes are significant, but do not warrant data recalibration

ii The row-level field for a table has changed by 10-50% compared to the `COMPARISON_FILE`

TRIVIAL

i Changes are insignificant (e.g., fixing typos; removing erroneous but unused rows from a table), and do not warrant data recalibration.

ii No changes made to the row or image

`PEDIGREE`

This should be `GROUND`, `DUMMY` or `INFLIGHT` (`MODEL` is accepted in some cases) and is populated with the value given in the header keyword of the FITS file. Note that this field should have been filled already by the CDBS script `mkload` for pipeline reference files and SYNPHOT throughput tables. In the case of the TMC file, however, this field is not filled by the `mkload` script. The value will have to be entered manually so that the PEDIGREE value for the TMC table matches the value in the SYNPHOT throughput tables that triggered the remake of the the TMC table.

`OBSERVATION_BEGIN_DATE` *and* `OBSERVATION_END_DATE`

These are the actual start and end date of acquisition of the calibration data used to create the reference file. The format should be Month Day Year (e.g., April 12, 2001) to be consistent with the `USEAFTER` date format. These fields are populated by the `mkload` script and are left blank when the `PEDIGREE` values are `DUMMY`, `GROUND`, or `MODEL`.

`COMMENT` *(in the row section)*

The `COMMENT` field in the row section can be blank if there are no relevant comments at the row level, but use of comments at the row level is strongly encouraged. We will be delivering reference file tables where only a few rows of the table have changed significantly as compared to the old reference file. In such cases, row-level comments may be more appropriate than header-level comments, and they are required under such circumstances.

An IRAF task called `setlodkeywd` has been developed for use in populating keyword fields in the "load" files automatically. This is particularly useful if a large number of files need to have fields populated in an identical manner. This task has been defined within the delivery account

IRAF tasks. An lpar of the task looks like this:

```
infile = "@filelist " File or list of lod files to fix
comments = yes Add comments?  (yes/no/append)
change_level = " " Change level value:  SEVERE, MODERATE, TRIVIAL
opus_flag = " " Opus flag (Y/N)
pedigree =" " Pedigree entry:  GROUND, DUMMY, IN-FLIGHT
(inlist = " ")
(inlod= " ")
(mode = " q")
```

where *infile* should have one "load" file name or a list of "load" files to be edited. You can create a list of "load" files with the command

```
ls *.lod >>& filelist
```

If a list of files is used, the '@' symbol has to precede the list name (as in the example). DO NOT use '*.lod' in the paramenter 'infile' because it won't work. Also, be aware that if you put the names of the FITS files here accidentally, this task will overwrite the .fits files and corrupt them, making it necessary to re-retrieve them from the deliverer's directory before you can continue.

comments = yes will copy `DESCRIP` and all `HISTORY` lines from the FITS reference file header, deleting what is currently present in this entry. If *comments = no*, nothing will be copied from the reference file and information present in this field will not be changed. Always set this to *comments = yes* unless the deliverer specifically indicated that the comment section of the file should have information different than that of the header of the FITS file.

change_level is the change level value. Refer to explanation above for the appropriate value to use. If it is left blank, the current entry will remain unchanged.

opus_flag = " " is the opus flag value. Refer to explanation above. If it is left blank, the current entry will remain unchanged.

pedigree = " " is the pedigree value. Refer to above explanation. If it is left blank, the current keyword value will be retained. Since the `mkload` task extracts this information from the header of the FITS file, we can leave this parameter blank.

inlist, inlod are list parameters used internally by the task. Do not enter any value here.

An example of a filled "load" file looks like this:

```
 FILE_NAME = 11x1_2001_1120_1125_ref_bia.fits

INSTRUMENT = stis
REFERENCE_FILE_TYPE = bia
USEAFTER_DATE = Nov 20 2001 00:00:00
COMPARISON_FILE= lbq12111o_bia.fits
OPUS_FLAG = Y
```

```
COMMENT = Superbias created by R. Diaz-Miller
Created on Dec 19, 2001 using the cl scripts
‘‘refbias’’ and "refaver", which are available
in the (local) xstis package within STSDAS.
Superbias image, combination of 98 input bias frames
taken in CCDGAIN=1, BINAXIS1=1, BINAXIS2=1 mode.
All input frames were from Proposal(s):
8901/8903 "CCD Bias Monitor".
The following input files were used:
o6hn2b010
o6hn2c010
o6hn2d010
o6hn2e010
o6hn2f010
o6hn2g010
cl script "refbias" was run on these input files,
after having split them up into sub-lists of less
than 30 imsets each.  After running "refbias" on the
individual sub-lists, script "refaver" was run to
average the reference files resulting from the
individual "refbias" runs together.
ENDHEADER

CHANGE_LEVEL = SEVERE
PEDIGREE = INFLIGHT
OBSERVATION_BEGIN_DATE = Nov 20 2001
OBSERVATION_END_DATE = Nov 25 2001
BINAXIS1 = 1
BINAXIS2 = 1
CCDAMP = D
CCDGAIN = 1
CCDOFFST = 3
DETECTOR = CCD
COMMENT =
ENDROW
ENDFILE
```

Note that in some cases a new reference table may be identical to its predecessor with the exception of some rows within the table. In this case, use the `CHANGE_LEVEL` from these rows as indicated in the delivery form and set to `TRIVIAL` the unchanged row groups. Currently, the task `setlodkeywd` can only change all of the lines in a “load” file to the same value. Should the file require multiple values, the “load” file will need to be edited “by hand” with your favorite text editor. An example of such a situation follows. A new PHT table where the G230LB mode was updated while the G230MB mode was unchanged has the following row sections:

```
CHANGE_LEVEL = SEVERE
PEDIGREE = INFLIGHT
OBSERVATION_BEGIN_DATE = May 21 1997
OBSERVATION_END_DATE = Jul 1 1997
CENWAVE = -1
DETECTOR = CCD
OBSTYPE = SPECTROSCOPIC
OPT_ELEM = G230LB
COMMENT = New calibration from program 9117
ENDROW

CHANGE_LEVEL = TRIVIAL
PEDIGREE = GROUND
OBSERVATION_BEGIN_DATE =
OBSERVATION_END_DATE =
CENWAVE = -1
DETECTOR = CCD
OBSTYPE = SPECTROSCOPIC
OPT_ELEM = G230MB
COMMENT =
ENDROW
```

Check the `COMPARISON_FILE` *parameter*

There are cases when the `mkload` script puts more than one entry in the `COMPARISON_FILE` parameter. In those cases, all the entries should be erased manually, except for one. Leave the most recent reference file from that list. If more than one file is in this parameter, the delivery will fail.

8. Certify the "load" files.

After creation of the "load" files they also need to be certified:

```
certify filename.lod >>& delivery.log
```

where "filename.lod" can be replaced by a wildcard (*.lod). If the reference FITS file, and consequently the "load" file, uses wildcard values, -1, -999, ANY, or N/A, for any of the header keywords (see ICD-47), `certify` will report the following "error" and the "load" file needs to be "expanded":

```
Error in opt_elem[1]:  ''any'' is not a legal value.  May need to run expload
```

```
Error in cenwave[1]: ‘‘-1’’ is not a legal value.  May need to run expload
```

Note that this applies to image reference files only, table reference files are expanded appropiately by the `mkload` script. With the term "expanded", we mean that the wild cards in the image "load" file have to be replaced with actual values. To "expand" the image "load" files, run the CDBS task `expload`. (The wild cards are usually more than one value and therefore the "expload" name.)

```
expload filename_in.lod filename_out.lod /store/smalls/cdbs/tools/data/####.rule
```

where the "*filename_in.lod*" file can be the same as "*filename_out.lod*"; in which case the changes will be written in the same file (note that this task does not take wildcard syntax on the command line). Expload expands the "load" file in those cases where a single reference file is to be used for many modes. For example, suppose we have a reference file that is applicable for ANY optical element (`OPT_ELEM`) of the STIS spectroscopic observing modes. `Expload` will "expand" the "load" file to cover all legal `OPT_ELEM` values, providing one row section in the "load" file for each of the `OPT_ELEM` values. The expansion of the files is governed by the so-called "####.rule" file, where #### is replaced by the instrument name. The rule files for each instrument are located in the CDBS data directory /store/smalls/cdbs/tools/data/ in the `smalls.stsci.edu` domain or /grp/hst/cdbs/tools/data/ for Solaris and Mac systems. This file shows the current legal values that will be used to replace wildcard values in the expansion. In the above example, the expanding rule for combination `OBSTYPE=SPECTROSCOPIC` and `OPT_ELEM=ANY` (taken from the stis.rule file) is:

```
OBSTYPE = SPECTROSCOPIC && OPT_ELEM = ANY =>
OPT_ELEM=G140L || OPT_ELEM=G140M || OPT_ELEM=E140M ||
OPT_ELEM=E140H || OPT_ELEM=G230L || OPT_ELEM=G230M ||
OPT_ELEM=E230M || OPT_ELEM=E230H || OPT_ELEM=PRISM ||
OPT_ELEM=G230LB || OPT_ELEM=G230MB || OPT_ELEM=G430L ||
OPT_ELEM=G430M || OPT_ELEM=G750L || OPT_ELEM=G750M ||
OPT_ELEM=X140H || OPT_ELEM=X140M || OPT_ELEM=X230H ||
OPT_ELEM=X230M;
```

That is, the row section with `OPT_ELEM = ANY` will be replaced by several row sections, one for `OPT_ELEM=G140L`, another for `OPT_ELEM=G140M`, etc. Once the file has been properly exploaded, run `certify` again until there are no errors or expansions required. Repeat this step as necessary until `certify` does not report any missing keyword information.

In order to simplify this work, one script has been created to expand several files of the same kind at once. The script is called `multi_expload` and is located in the bin/ directory of the delivery account in SMALLS. This script needs the information of the instrument to which these files apply and the extension of the file (in this case it is the last characters of the file name and not the type of reference file). For example, to run this command for ACS dark reference files with names "*drk_new.fits", type:

```
multi_expload drk_new acs
```

Note that you first have to give the extension of the file and then the instrument name. If you are delivering files with different extensions, you have to run this script for each extension.

9. Run check_load

The CDBS task, `check_load`, must be run on the "load" files before they can be delivered. This task takes wildcards.

```
check_load *.lod >>& delivery.log
```

The output from this task will look like the following:

```
starting check_load
database cdbs_ops
server CATLOG
Thu Feb 26 11:25:26 EST 1998
load file:  i2916173o_drk.lod
header file:  i2916173o_drk.fits
Wrote file (i2916173o_drk.lod)
no differences for file i2916173o_drk.lod
```

10. Rename the files to have an unique name identifier

Once the reference files are ready to be ingested into the databases, the files have to be renamed with an unique name identifier.

SYNPHOT

In the case of SYNPHOT tables an incremental number format is used for the naming of new data files. When the SYNPHOT files were delivered they are likely to have been already re-named to the next value; however, make sure that this is the case by checking the SYNPHOT disk area (/grp/hst/cdbs/comp/ in the centralized storage area or /store/smalls/ref/ in the `smalls.stsci.edu` domain). Identify the type of file by its name and check that indeed the number of the new file does not exist. In some cases, the instrument teams choose to skip values. This is not a problem and you can deliver the files that way provided that the number does not previously exist and it is greater than that of any previously delivered file of that type. If these files do not have an unique name, however, use the task `uniqname` as described for the pipeline reference files. Be sure in this case that the assigned number is larger than that of the older files. We have found instances where `uniqname` assigns a value of "001" when this type of file has been previously delivered many times.

Note that if SYNPHOT Atlas files (e.g. Kurucz) and HST Calibration Standards spectra are delivered, those have special names that cannot be changed and therefore the same file name should be used to replace the old file.

Pipeline Reference Files

For the case of calibration reference files, assign an unique name using the CDBS script called `uniqname`:

```
uniqname *.lod >>& delivery.log
```

The files will be renamed to CDBS style reference file names. More details about the naming conventions used by the script are described in the CDBS documentation web page (http://www.stsci.edu/instruments/observatory/cdbs/documents/).

Note that the WFPC2 team usually renames the files themselves. This is because, at least for bias and darks reference files, they use an automatic script that does this step. In these cases we don't have to run `uniqname` on the files .

11. SYNPHOT Bandpases

Non HST bandpasses files are a special case of SYNPHOT files. These are delivered to CDBS and are part of the SYNPHOT package. Unlike the atlases, these are in the CDBS database and as such should be included in the TMC table. Examples of these files are the Landolt b, i, r, u, v filter bandpasses and the Stromgren B, U, V, Y. These files are located in the directory /grp/hst/cdbs/comp/nonhst/ and follow the same numbering convention as any of the SYNPHOT files. Given that those files are non HST filters and used for normalization by all the HST instruments, these should in practice be tested by all the instrument teams.

If the person making the update or requesting it has not communicated about this change to all the instrument teams, we should make sure we let all the instrument teams know the reason behind this change, the person making the change, the location of these file and the timeline for testing and sign-off for the delivery of these files. The instrument teams have to test SYNPHOT and if possible the ETCs, before this file can be delivered. The test procedures should be left to the instrument teams; however, if this work is done by only one person familiar with the use of these files with SYNPHOT, the instrument teams should agree that this person will be the tester and sign-off to this effect via an e-mail to the INS/CDBS team. If the instrument teams decide to perform the test themselves, they should sign-off the files for delivery once testing is completed. These files cannot be delivered until sign-off from all the instrument teams is obtained. There might be cases where there is a pressing need to deliver these files and therefore it is important to make sure a deadline is set for the delivery of these files. This deadline should be clearly given to the instrument teams when letting them know about the change in the files. If any of the teams has a problem with meeting this deadline, contact the CDBS Lead for assistance in working out an schedule with them.

12. Put the files in a delivery directory and deliver them to the DMS

In the case of pipeline reference files or SYNPHOT Throughput tables, copy the FITS and "load" files to be delivered to an empty directory. The delivery script does not work if there are other files in this directory. Some empty directories already exist for this purpose. These are under /calib/cdbs_delivery/ directory and have names "deliverfiles*". Deliver the reference files to the CDBS database, and when applicable to the OPUS and DADS databases, using the `sendit` script:

```
sendit >>& ../workingdir/delivery.log
```

where the "workingdir" is the delivery path where the files were tested before delivery. This script will re-check that the files are FITS and CDBS compliant, will create SQL command inputs for the databases, and will copy the delivered files to a fixed location from where the Data Management System Teams will collect the data.

In the case of WFPC2 files, `sendit` converts the GEIS files to "waiver" FITS type before they are delivered to the databases or DMS disks. In the particular case of SYNPHOT Atlas files (e.g. Kurucz), the files cannot be delivered this way so skip this step.

More detailed information on the steps performed by `sendit` will be described in another TIR. An example of the `sendit` output for a successful delivery is:

You start your delivery process at:Mon Apr 4 19:39:02 GMT 2005
starting deliver_cdbs
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:02 GMT 2005

---

starting certify_delivery
Mon Apr 4 19:39:02 GMT 2005
== Checking p441909no_pht.fits ==
== Checking p441909no_pht.lod ==
certify_delivery succeeded

---

starting loopfits_delivery
Mon Apr 4 19:39:03 GMT 2005
converting:
created output file loopfits.out
loopfits_delivery succeeded

---

starting farris_fitsverify_delivery

Mon Apr 4 19:39:03 GMT 2005
no errors or warnings reported by farris_fitsverify
created output file farris_fitsverify.out
fitsverify_delivery succeeded

---

starting check_load
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:03 GMT 2005
load file: p441909no_pht.lod
header file: p441909no_pht.fits
Wrote file (p441909no_pht.lod)
no differences for file p441909no_pht.lod
lcheck_load succeeded

---

starting cdbs_sql_gen
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:05 GMT 2005
delivery_number = 11560
lock acquired
load file(s) p441909no_pht.lod
Processing p441909no_pht.lod ...

Warning: No comparison file records matched mode values for row 6.
New equivalence class values and a SEVERE change_level were used

Processing complete – cdbs_delivery11560.sql generated
cdbs_sql_gen succeeded

---

starting run_delivery_sql
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:08 GMT 2005
/calib/cdbs_delivery/deliverfiles2/cdbs_delivery11560.sql.out
using file /calib/cdbs_delivery/deliverfiles2/cdbs_delivery11560.sql
no errors in processing sql file /calib/cdbs_delivery/deliverfiles2/cdbs_delivery11560.sql
created output file /calib/cdbs_delivery/deliverfiles2/cdbs_delivery11560.sql.out
run_delivery_sql succeeded

---

starting check_cdbs
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:09 GMT 2005
delivery 11560 in progress

missing modes check
uni check
synphot compname check
expansion number check
archive date check
general availability date check
opus load date check 1
opus load date check 2
row check
file check
reject check 1
reject check 2
current reject check 3
current delivery number check
reject check 4

output file check_cdbs_11560.out created
4 warning(s): see output file check_cdbs_11560.out
No errors.
check_cdbs succeeded

---

starting opus_sql_gen
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:39 GMT 2005

created opus_11560_o.sql
opus_sql_gen succeeded

---

starting update_ga_date
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:39 GMT 2005
delivery number = 11560
general availability date was updated
cdbs_ops
lock released
update_ga_date succeeded

---

starting opus_catalog
database cdbs_ops
server CATLOG
Mon Apr 4 19:39:40 GMT 2005
delivery_number= 11560
instr= o

catalog file= opus_11560_o.cat
opus_catalog succeeded

---

deliver_cdbs completed
Mon Apr 4 19:39:41 GMT 2005

total execution times:

real 39.1
user 7.0
sys 7.4

#####

...CDBS process done...Making links for delivery pick-up... linking p441909no_pht.fits
linking opus_11560_o.cat
linking opus_11560_o.sql

#####

...You have successfully finished the delivery process...

#####

Process finished at:Mon Apr 4 19:39:42 GMT 2005

###
###
###
###

The `sendit` script performs the basic tests and creates SQL command input files. After these are completed, a unique delivery number is assigned (indicated in bold in the above output). If the delivery happens to fail after this number has been assigned, the delivery has to be cancelled before another delivery or redelivery can occur. This is done running the command `delete_delivery`:

```
delete_delivery >>& ../workingdir/delivery.log
```

This will unlock the databases and will correctly exit the delivery process. An example of a failed delivery is:

```
run_delivery_sql succeeded
——————————————————————
starting check_cdbs
database cdbs_ops
server CATLOG
Mon Apr 4 19:16:25 GMT 2005
delivery 11559 in progress
missing modes check
uni check
synphot compname check
expansion number check
archive date check
general availability date check
opus load date check 1
opus load date check 2
row check
file check
reject check 1
reject check 2
current reject check 3
current delivery number check
reject check 4

output file check_cdbs_11559.out created
4 warning(s): see output file check_cdbs_11559.out
1 error(s): see output file check_cdbs_11559.out
CDBS ERROR: check_cdbs failure. Exiting.

real 33.7
user 6.4
sys 5.3
FAILURE of deliver_cdbs.
```

In this example, the first successful lines of the `sendit`'s output are not shown. If the delivery fails before the delivery number has been assigned, the `delete_delivery` command does not need to be run.

13. Fill the delivery form and e-mail it to DMS.

If the `*.lod` files have `OPUS_FLAG = Y` (e.g. all pipeline reference files), immediately after the delivery software (or `sendit`) successfully populated the CDBS database, you have to notify

DMS of the delivery; so they can check that the files are ingested properly in the different DMS areas. When `OPUS_FLAG = N`, the files are sent only to the CDBS database, as those do not affect the pipeline calibration products and should only be used by SYNPHOT. In the latter case DMS should not be notified. For those cases when a notification is needed, submit an e-mail to the e-mail address **cdbs_datamng@stsci.edu** using the following formatted form (a template of this form is in the file /calib/cdbs_directory/form):

```
Date:
By:
Instrument:
File_type(s) (e.g.  PHT, DRK):
Directory where data is found:
/calib/cdbs_delivery/.../2005_..._.../

Description of data delivered:


Delivery_number:

Opus ingest date:
Opus signoff
```

where *Date* is today's date. In *By:*, put your name; in *Instrument*, put the name of the instrument or team for which you are delivering the reference files; in *File_type(s),* put the extension (e.g. PHT, DRK) of the file delivered. The information in *Directory where data is found:* is for our records. This directory is the directory where you tested the files before delivery, i.e. your working directory. In this case replace the "..." by the appropriate values according to the instrument and the date of delivery. In the section *Description of data delivered:* list the datasets delivered and the *opus** ASCII files that were created by the script `sendit`. For example, use the `ls -la`:

```
-rw-rw-rwx 4 srefpipe 108 Mar 25 16:56 opus_11556_j.cat*
-rw-rw-rwx 4 srefpipe 1499 Mar 25 16:56 opus_11556_j.sql*
-rw-rw-rwx 4 srefpipe 164160 Mar 25 16:53 p3p1650tj_mdz.fits*
-rw-r--r-- 1 srefpipe 1046 Mar 25 16:53 p3p1650tj_mdz.lod
```

and copy and paste this information to the delivery form. In the case of WFPC2 files, list only the "waiver" FITS files and "opus*" files. Finally, in *Delivery_number* put the number of the delivery. The subject of this e-mail has to be: *Delivery #####*, where ##### is the number of the delivery. The last two fields (opus ingest date and opus ingest signoff) are left blank and will be filled by the OPUS team's person ingesting the file. Save a copy of the completed form in the directory `/calib/cdbs_directory/deliveryforms/` with a filename of the format *yyyy_mm_dd_ins* where ins is the instrument, such as acs, wfpc2, etc.

14. Transfer the files to the centralized storage area

Currently the deliveries are made in the SunFire15K system and these disks cannot be mounted in the Science Cluster. Therefore, a copy of the reference files have to be transfered to the directories located in the centralized storage area. In this area, each of the instrument teams has an assigned area to store the reference files, one for pipeline reference files and another for SYNPHOT reference files. Each of these directories can be accessed from the /grp/hst/cdbs/ directory. The specific disk location for each of the instrument teams is:

	Directory Path	
	reference files /grp/hst/cdbs/	
Instrument	calibration	SYNPHOT
ACS	jref/	comp/acs/
STIS	oref/	comp/stis/
WFPC2	uref/	comp/wfpc2/
NICMOS	nref/	comp/nicmos/
WFC3	iref/	comp/wfc3/
COS	lref/	comp/cos/
non-HST	-	comp/nonhst/
OTA	-	comp/ota/
TMC/TMG	-	mtab/
Atlases	-	grid/
CALSPEC	calspec/	current_calspec/

In the case of WFPC2, transfer both the GEIS and "waiver" FITS files to the centralized storage area. With some types of WFPC2 files, there is a second set of files to be used only on linux systems. This will be indicated in the delivery form sent by the WFPC2 team. The linux version of the files will be in a separate directory to be retrieved by the CDBS team. The CDBS deliverer will download the linux files to a linux directory within the delivery directory, but the linux files do not go through the delivery process with the Solaris files. Once the Solaris files have been delivered, the GEIS linux files (not the lod files) will need to be transferred to their own directory, `/grp/hst/cdbs/uref_linux/`.

In the case of SYNPHOT data files, this step should be done before creating the TMC file with the `MKCOMPTAB` task, as the software looks in the above mentioned disk location for the files that appear as active in the CDBS database. If the files are not present in these disk locations, the task will fail. Currently, this task runs in `smalls`, however, we need to copy these throughput files to the `smalls` reference files area (`/store/smalls/ref/thu/`) because these are not copied automatically by the "sendit" script.

In the case of SYNPHOT Atlases, FTP the files to their corresponding `grid` directory in the centralized storage area. This will make them accessible to all internal users. In order for these atlases to be used by SYNPHOT outside the Science Cluster (including MAC users), users have to copy them to their corresponding /grid/ directories (where all the SYNPHOT Atlases and Libraries live) or wait for the next release where the files will be automatically delivered to them with the SYNPHOT package. Note that copying them to the /grp/hst/cdbs/grid/ directory in the centralized storage area makes these files available to the STSDAS Group for future STSDAS releases or for download from their web site. In any case, **it is necessary to notify the STSDAS Group** that these Atlases or the HST Calibration Standards spectra have changed so they can package them in the appropriate tarball. See Apendix C for further information regarding the needed test and delivery procedures for these files.

To transfer the files, SFTP or FTP to the centralized storage area. Use the **srefpipe** account name and password. Change the transfer mode to *binary* (if using FTP) and put in all the FITS files (GEIS and "waiver" FITS in the case of WFPC2), into the respective directory. For example,

```
mymac> sftp srefpipe@tib.stci.edu
Connecting to tib.stsci.edu...
Password:
sftp> cd /grp/hst/cdbs/jref/
sftp> mput *.fits
```

15. In the case of SYNPHOT files: Create the TMC table and deliver it with the TMG and TMT tables

In the case of SYNPHOT data files, once these have been delivered, it is necessary to re-create the TMC table using the CDBS script `mkcomptab`. In your testing directory or a subdirectory created just for the TMC, TMG and TMT tables, run:

```
mkcomptab new_tmc.fits
```

This script will recreate the TMC table using the information located in the CDBS database. It will use the most up to date files to fill each of the `COMPONENT` rows in this file. However, if it encounters more than one file with the same `USEAFTER` date, it will list all of them in the TMC table. The MKCOMPTAB task will list these files from older to newer. This order is correct for SYNPHOT, however, Pysynhot uses the files in the opposite direction. Currently, Pysynphot supports only some COS calculations (in order to support the ETCs), however, we are in the process of transitioning from SYNPHOT to Pysynphot and therefore the TMC files should not have repeated entries, to avoid a conflict between Pysynphot and SYNPHOT. In order to prevent repeated entries the USEAFTER date has to be different for each of the files delivered. This is why it is important to set the USEAFTER date of SYNPHOT throughput tables to the date when the file is created or delivered. Make sure this is the case when you check the changes in this file in order to prevent repeated entries. More details about this script can be found in the CDBS Documentation web page (http://www.stsci.edu/instruments/observatory/cdbs/documents/).

Once the TMC file has been created, check that the changes in this file are for those SYNPHOT data files just delivered. For this, run the IDL procedure compare_table.pro. The script is located in the Science Cluster in the STScI IDL area (/data/garnet2/idl/stsci/) and in the smalls area (/store/smalls/srefpipe/useful_scripts/). Keep in mind that to run IDL in smalls, you must go into the .setenv file for smalls (srefpipe account) and comment out the line that says "source ~/def/opus_login.csh", save the file, and open a new window that will then run IDL. To compare the old and new TMC tables type in IDL:

```
IDL> .compile /store/smalls/srefpipe/useful_scripts/compare_table.pro

IDL> compare_table,'path1/new_tmc.fits','path2/old_tmc.fits',$
COLUMNS=['compname','filename'],SAVEFILE=1
```

The older TMC tables are located in /store/smalls/ref/mul, and the current version of the TMC table is the last one listed. The COLUMNS parameter indicates which set of columns to use for the comparison of each row of the file. When the SAVEFILE parameter is set equal to 1, it will direct the output of the procedure to a file called compare_table.out in the current directory. This script looks for missing elements in the table by checking differences in each row. If no unexpected differences are found, fill the HISTORY section of the FITS file documenting the reason for the file to be re-created.

If the TMG and/or TMT tables were received together with the SYNPHOT data files, copy these files to the directory where the TMC file was created and check that the header keywords are correctly populated. Also check that the changes in the file are as expected. For this, use the same IDL procedure compare_table.pro. For the TMT table the COLUMNS parameters are set identical to those used for the TMC table, while for the TMG table these should be ['compname','keyword','innode','outnode','thcompname']. We do expect the deliverer to have done this test already; but we have to confirm the changes.

If no problems or unexpected differences are found, fitsverify and certify the TMG, TMT, and TMC files; as in steps 4 and 5. Create the "load" files as in step 6. Fill the fields CHANGE_LEVEL and PEDIGREE of the TMC "load" file value used by the throughput tables that were just delivered. That is, if the throughput tables had CHANGE_LEVEL = SEVERE use this value for the CHANGE_LEVEL of the TMC "load" file. For the TMG and TMT "load" files always use SEVERE. For these three files use OPUS_FLAG=Y. Run certify, check_load, and uniqname on the "load" files according to steps 8, 9 and 10. Before these files can be delivered, however, they must be tested against the SSB suite of regression tests. To accomplish this, the current procedure is to place the TMC, TMG and the latest TMT file in /grp/hst/cdbs/work/vicki/ for Vickie Laidler and e-mail her so that she knows where the files are and that they need to be tested. Note that if the TMT file and TMG files were not delivered, you need to find the latest delivery files and put them in Vicki's directory together with the new TMC file. This is because she needs these files to be in the same directory to do her test. Only after receiving an e-mail from her confirming that the files are ready for delivery should the files be delivered as described in step 11. Then send a delivery form to DMS as in step 12, and finally transfer the files to centralized storage as in step 13 and copy only the new file(s) to the /store/smalls/ref/mul/ directory.

16. Run cdbs_report

After the CDBS delivery Pipeline completes all the stages succesfully, an e-mail acknowledging the completion of the delivery is sent back to the INS/CDBS member delivering the files. (A copy is sent to the cdbs@stsci.edu e-mail address.) This usually happens the same day the files were delivered. Note that in the case when the files were delivered late in the day, the acknowledgment of the ingest will arrive the next day. If the reply e-mail is not received within the expected time, investigate the reason for the delay. The OPUS team usually notifies us of the successful ingestion after the files have been properly transferred to the Archive, OPUS, and the mirror sites (ECF and CADC) disks. Although a problem in any of these steps can delay the notification, after the files have been ingested into the Archive and OPUS disk areas, the files will be used in the OTFR pipeline and will be available for retrieval. But before these files can be recommended as the best reference files for a given dataset, it is necessary to run another script that updates the archive database. The *_ref_data tables in the archive database are used to select the best reference files via the "Best Reference Files" option in the archive retrieval form. The script that updates these tables is run only once a day (usually at night) and therefore there is a period of time when the files used in OTFR are different than those selected by the "Best Reference Files" option. This does not affect our delivery, but it is something to keep in mind. In any case, once OPUS has ingested the files, we can assume the files are in the system. The information on when the files were ingested into the Archive and OPUS system can be obtained running the `cdbs_report` script:

```
cdbs_report #####
```

where ##### is the delivery number. For example, running this script for delivery number 11160 shows all the information relevant to that delivery number:

CDBS Installation Report

Instrument	Reference File Type	File Name	Useafter Date	Archive Date	OPUS Load Date
STIS	PHT	P441909NO_PHT.FIT	Mar 15 1999 12:00AM	Apr 4 2005 8:32PM	Apr 4 2005 8:32PM
STIS	PHT	P441909OO_PHT.FIT	Oct 1 1996 12:00AM	Apr 4 2005 8:32PM	Apr 4 2005 8:32PM
STIS	TDS	P441909PO_TDS.FIT	Oct 1 1996 12:00AM	Apr 4 2005 8:32PM	Apr 4 2005 8:32PM

17. Check the size of the files in the archive

We have found several cases where the tables that are ingested into the archive are corrupted. We believe that this happened when the files were ftped to the archive media. Since we are now delivering the files from the SunFire15K system, where the operational and archive domains reside, this problem is likely to have been solved; however, we should still perform this check to verify the integrity of the files that are being archived. This can be done by comparing the size of the archived reference files to that of the files we have in our delivery directory. This verification can be done in three different ways, all by performing a query to the *archive_files* table in the database **dadsops** in the ZEPPO server.

The simplest way is by running the CSHELL script `search_size_csh` . This script was created to check the size of the files and is available in the special CDBS account only. It makes the appropriate calls to the database using the provided dataset name search string. To run this script type:

```
smalls> search_size_csh NNNNN
```

where `NNNN` should be the file name or file name prefix to search. Note it has to be entered as caps. If the file prefix is not provided, the script will request it. The output to this script is a file called `size_out.txt` which has the commands and output of the SQL database search.

If the script is not available, you have to perform each step manually. The first two steps would be to set the environment variable `DSQUERY` and load the *dadsops* database. For this, type:

```
mymac>setenv DSQUERY ZEPPO
mymac>isql
1>use dadsops
2>go
```

The column we want to search here is: *afi_data_set_name*. The rows we want to examine are those that have values equal to the name of the reference file we just delivered. For example, if we just delivered reference files with names p441909so_drk.fits and p441909to_bia.fits, we can check the size of the files in the archive with the command:

```
1>select * from archive_files where afi_data_set_name like ‘‘P441909%’’
2>go
```

where "%" is a wild card. In this case, using the wild card will simplify the verification by showing us the size of all the reference files with prefix `"P441909\%"`. Note that we are using uppercase for the file name. This is because the names of the reference files are stored as uppercase. The output of this command looks like this:

```
     afi_data_set_name          afi_archive_class
afi_generation_date       afi_mission afi_file_extension
afi_file_name
afi_file_type afi_pre_compress_size afi_post_compress_size
afi_checksum   afi_verify_status    afi_virtual
-------------------------------------- -----------------
------------------------- ----------- --------------------
---------------------------------------------------------------------------------------------
```

```
------------- --------------------- ----------------------
------------ ----------------- -----------
P441909NO            CTB
Apr 4 2005 8:06PM    HST     PHT
p441909no_pht.fits
FITS         4518720.000000     2700547.000000
1336725531   NULL      N
P441909OO             CTB
Apr 4 2005 8:06PM   HST     PHT
p441909oo_pht.fits
FITS         4518720.000000     2700418.000000
759764626   NULL      N
```

To simplify the output you could select the *afi_pre_compress_size* column only. For this the command should be:

```
1>select afi_data_set_name,afi_pre_compress_size
2> from archive_files where afi_data_set_name like "P441909%"
2>go
```

The output will look like this:

```
     afi_data_set_name     afi_pre_compress_size
--------------------------------------- -------------
P441909NO     4518720.000000
P441909OO     4518720.000000
```

Another way to do this is by entering the SQL commands listed above in an ASCII file; for the example used here the file is called `size_query.sql`. Once the file has been created, run the following in the command:

```
smalls> isql -e -i size_querry.sql -o size_out.txt -S ZEPPO
```

the output will be directed to a file called `size_out.txt`.

In all cases, the last step is to compare the *afi_pre_compress_size* column value with the size of the file you have in your delivery directory. If these values are not identical, it is likely that the file in the archive is corrupted and the OPUS team has to be informed of the problem.

18. Verify the correct usage of the reference files in the operational environment.

Another problem that we have seen in the past has to do with the way the new reference files were recommended. In a few cases, the files were not ingested properly and the old reference file was still being recommended for some datasets when it should not have been. Therefore, it was decided to verify that the reference files were used correctly by OPUS and properly recommended in the archive. This should be done only after the nightly script that updates the *ref_data tables, containing information about the Best reference files has completed. To make sure that this has run, wait until the day after the files were delivered to CDBS to perform this test. Note that since there is no science data for COS or WFC3 yet, this step should be skipped for these instruments until after SM4. When HST starts to take science observations with these instruments, their *ref_data* tables will start to get entries and therefore this step can be performed.

For the other instruments, the simplest way to do this is by running the CSHELL script `search_best_reffile`. This script was created to facilitate verification about the usage of the reference files. Currently, it only supports ACS, WFPC2, and STIS reference files. This script sends the request to the DADSOPS database for the best reference files used after a given date. It returns the list of best reference files used for a particular instrument, detector, reference file type, and time of observation. Just like the script that checks the size of the reference files, this script can only be run in the special CDBS account.

This script can be run as follows:

```
smalls> search_best_reffile INSTRUMENT TYPE_FILE USEAFTERDATE DETECTOR
```

where INSTRUMENT is the INSTRUMENT name for which the search is made. TYPE_FILE is the type of reference file we want to use in the search; e.g. DRK, BIA, PHT, or drk, bia, pht. USEAFTERDATE is the date after which we want to verify the usage of the reference files. In this case, it should correspond to the earliest USEAFTER date of the reference file type we are delivering. The format for the USEAFTERDATE is as follows: Month Day Year Time. Finally, DETECTOR is the name of the detector for which this file applies. If there is no detector distinction for the reference files (e.g. WFPC2 DRK reference files), this parameter can be left blank.

For the moment, this script does not seem to work in an xterm or xgterm. You may need to open a basic terminal in order for the script to work. An example of using this script to find WPFC2 darks is:

```
smalls> search_best_reffile WFPC2 DRK Jan 21 2008 04:10:00
```

If the script is not available, you will have to perform each step manually. In order to do this, first we need to know the prefix of the field name, in the tables with reference file data, that is associated to the reference files. That is, the reference file records are located in tables named "###_ref_data"; where ### is the name of the instrument (e.g. acs_ref_data). Within these tables the reference files are listed in columns named after the calibration reference file name that appears in the header of the observation FITS files. For example, the "Pixel to pixel flat field" file for ACS data is assigned by the keyword PFLTFILE. The column in the *acs_ref_data* that contains this information is *acr_best_pfltfile*. Note the prefix used for the reference file column, these change from instrument to instrument but are the same for all the reference files of that instrument. Here is the list of prefixes :

ZEPPO database table ###_ref_data	
Instrument	prefix
ACS	acr_best_###
STIS	ssr_best_###
WFPC2	w2r_best_###
NICMOS	nsr_best_###

where ### is the reference file table identifier. In appendix A of this document the corresponding names for the current reference files for all the instruments are listed. The DSQUERY environment variable as well as the database should be set as in step 17. Once the table identifier is known, the verification can be done using the SQL command:

`select distinct` *reference_file_column* `from` *instrument_best_ref_table* `where` *prefix_expstart_field* `>= "`*USEAFTER_date*`"`

where for the above example *reference_file_column* is "acr_best_pfltfile" and *instrument_best_ref_table* is "acs_ref_data" The field *prefix_expstart_field* is the table column name with the information of the useafter date. In the case of ACS , *prefix_expstart_field* should be replaced by "acr_expstart", while for STIS it is "ssr_texpstrt" (see Appendix A for a complete list of all the parameters). Finally, "*USEAFTER_date*" is the useafter date reference file header keyword and after which the reference file has to be used; e.g., "MAR 05 2005 08:44:17". Note that we are using uppercase letters and specifying the hour, minute and second after which the file should be used. If uppercase does not work, use the same format as the file that was delivered. An example of the command to check the ACS darks recommended after the useafter date "Mar 05 2005 08:44:17" is:

```
1>select distinct acr_best_darkfile from acs_ref_data where acr_expstart >= "MAR 05
2005 08:44:17"
2> go
acr_best_darkfile
------------------
NULL
P3V2228OJ_DRK.FITS
P3V2228PJ_DRK.FITS
P3V2228QJ_DRK.FITS
```

This can be done by querying for the best reference files of any one kind in the **dadsops** database in the ZEPPO server. This command should list only the reference files that are active. Those that have been superseded by the current delivery should not appear in the list. If any of the old reference files appears (this means that there was a problem with the script that updates the *_ref_data tables), contact Mike Swam so he can re-run the cron job that updates this table. If possible, look for some examples of data have the erroneous reference file. For the latter you can use the StarView forms that list the best reference files; in those forms search for the old reference file in the corresponding field. Note also that in the above output there is a *"NULL"* value. When datasets are ingested in the archive, the best reference values are all set to *"NULL"*. This value is automatically changed later to the appropriate reference file value when the nightly cron job, that updates the best reference files tables, runs.

19. Send notification

Forward the acknowledgment e-mail from OPUS mentioned in step 15 to the deliverer along with a copy of the CDBS installation report (cdbs_report). This will serve as a confirmation that the files are in the system. Copy the "opus_*" files created by `sendit` to the testing directory and compress the files. Once the "opus_*" files are copied to the testing directory, delete all the files from the delivery directory, ie deliverfiles, deliverfiles2, deliverfiles3, etc, so that the next deliverer can find an empty directory in which to deliver their files.

20. Fill out delivery information on WIKI

Once the delivery is finished, the deliverer should report the completed delivery on the CDBS WIKI page. The main WIKI page for CDBS is located at http://www.stsci.edu/wiki/INS-CDBS/CDBSGroupNotes . From the main page, go to the link 'Status Reference File Deliveries'. On this page, you will find a table with the following headers; Date, Instrument, Delivery Number, Number of files delivered, Type of files delivered, and Name of Deliverer. Click on the edit button to add your current information. When the edit screen shows up, copy the format (html tags) of the previous line and insert your information into the html formatting. When you are done, you can Preview or Save the information.

This Wiki page will help all the members of the CDBS team keep informed as to how many and what types of deliveries are being made.

References

C. Cox, & C. Tullos TIR OSG-CAL-97-02 (updated 7/1/98)

R. Diaz-Miller TIR CDBS 2005-02

Appendix A

ACS

Table A1: acs_ref_data table reference useful keywords

Column_name	comment
acr_aperture	Aperture Name
acr_ccdamp	CCD Amplifier Readout Configuration
acr_ccdchip	CCD chip
acr_ccdgain	Commanded gain of CCD
acr_crsplit	number of cosmic ray split exposures
acr_detector	Detector
acr_expstart	UT date of start of observation (MMM DD YYYY hh:mm:ss)
acr_filter1	element selected from filter wheel 1
acr_filter2	element selected from filter wheel 2
acr_flashcur	Post flash current: OFF, LOW, MED, HIGH
acr_fwoffset	computed filter wheel offset
acr_fwerror	filter wheel position error flag: F or T
acr_obstype	Observation type - imaging or spectroscopic
acr_proposid	PEP proposal identifier
acr_shutrpos	Shutter position: A or B
acr_sclamp	lamp status, NONE or name of lamp which is on

Table A2: acs_ref_data table reference file identifier

Column_name	Reference file type	Reference file extension
acr_best_biasfile	BIAS IMAGE	BIA
acr_best_cfltfile	CORONAGRAPHIC SPOT FLAT IMAGE	CFL
acr_best_darkfile	DARK IMAGE	DRK
acr_best_dfltfile	DELTA FLAT IMAGE	DFL
acr_best_dgeofile	GEOMETRIC DELTA IMAGE (DISTORTION)	DXY
acr_best_flshfile	POST FLASH IMAGE	FLS
acr_best_lfltfile	LOW ORDER FLAT IMAGE	LFL
acr_best_pfltfile	PIXEL TO PIXEL FLAT FIELD IMAGE	PFL
acr_best_shadfile	SHUTTER SHADING IMAGE	SHD
acr_best_atodtab	ANALOG-TO-DIGITAL TABLE	A2D
acr_best_bpixtab	BAD PIXEL TABLE	BPX
acr_best_ccdtab	CCD PARAMETERS TABLE	CCD

Table A2: acs_ref_data table reference file identifier (cont)

Column_name	Reference file type	Reference file extension
acr_best_comptab	THE HST MASTER COMPONENT TABLE	TMC
acr_best_crrejtab	COSMIC RAY REJECTION PARAMETER TABLE	CRR
acr_best_graphtab	THE HST GRAPH TABLE	TMG
acr_best_idctab	IMAGE DISTORTION COEFFICIENTS TABLE	IDC
acr_best_mdriztab	MULTIDRIZZLE PARAMETER TABLE	MDZ
acr_best_mlintab	MAMA LINEARITY TABLE	LIN
acr_best_oscntab	CCD OVERSCAN REGION TABLE	OSC
acr_best_phottab	PHOTOMETRY and THROUGHPUT TABLE	PHT
acr_best_spottab	SPOT POSITION TABLE	CSP

STIS

Table A3: stis_ref_data table reference useful keywords

Column_name	comment
ssr_aperture	Aperture name
ssr_binaxis1	axis1 data bin size in unbinned detector pixels
ssr_binaxis2	axis2 data bin size in unbinned detector pixels
ssr_ccdamp	CCD Amplifier
ssr_ccdgain	CCD commanded Gain
ssr_ccdoffst	Commanded bias offset of CCD
ssr_cenwave	Central wavelength in Angstroms
ssr_crsplit	Number of CR split exposures
ssr_detector	Detector
ssr_lampset	spectral cal lamp current value (milliamps)
ssr_obstype	Observation Type (Imaging or Spectroscopic)
ssr_opt_elem	Optical Element used for observation
ssr_texpstrt	UT time of the start of exposure (MMM DD YYY hh:mm:ss)
ssr_wavecal	wavecal image file name

Table A4: stis_ref_data table reference file identifier

Column_name	Reference file type	Reference file extension
ssr_best_biasfile	Bias image file	BIA
ssr_best_darkfile	Dark image file	DRK
ssr_best_pfltfile	Pixel-to-pixel flat file	PFL
ssr_best_dfltfile	Delta flat image file	DFL
ssr_best_lfltfile	Low-order flat image file	LFL
ssr_best_shadfile	Shutter shading correction image file	SSC
ssr_best_sdstfile	Small scale distortion image file	SSD
ssr_best_atodtab	A2D Correction Table	A2D
ssr_best_apdstab	Aperture Description Table	APD
ssr_best_apertab	Aperture Throughput Table	APT
ssr_best_bpixtab	Bad Pixel Table	BPX
ssr_best_ccdtab	CCD Parameters Table	CCD
ssr_best_crrejtab	Cosmic Ray Rejection Parameters Table	CRR
ssr_best_disptab	Dispersion Coefficients Table	DSP
ssr_best_inangtab	Incidence Angle Correction Table	IAC
ssr_best_idctab	Image Distortion Correction Table	IDC
ssr_best_mlintab	MAMA Linearity Table	LIN
ssr_best_lamptab	Calibration Lamp Table	LMP
ssr_best_mofftab	MAMA Offset Correction Table	MOC
ssr_best_pctab	Photometric Correction Table	PCT
ssr_best_phottab	Photometric Conversion Table	PHOT
ssr_best_sdctab	2-D Spectrum Distortion Correction	SDC
ssr_best_cdstab	Cross-Disperser Scattering Table	CDS
ssr_best_echsctab	Echelle Scattering Table	ECH
ssr_best_ecstab	Echelle Cross-Dispersion Scattering Table	EXS
ssr_best_halotab	Detectore Halo table	HAL
ssr_best_riptab	Echelle Ripple Table	RIP
ssr_best_srwtab	Scattering reference Wavelenght Table	SRW
ssr_best_psftab	Telescope Point Spread Function Table	TEL
ssr_best_tdctab	NUV Dark Correction Table	TDC

Table A4: stis_ref_data table reference file identifier (cont)

Column_name	Reference file type	Reference file extension
ssr_best_tdstab	Time Dependent Sensitivity Table	TDS
ssr_best_wcptab	Wavecal Parameters Table	WCP
ssr_best_sptrctab	1-D Spectrum Trace Table	1DT
ssr_best_xtractab	1-D Extraction Parameters Table	1DX

WFPC2

Table A5: wfpc2_ref_data table reference file identifier

Column_name	Reference file type	Reference file extension (type)
w2r_best_atodfile	Analog to Digital Converter Lookup Table	R1?
w2r_best_biasfile	Bias Frame	R2?
w2r_best_darkfile	Dark Frame	R3?
w2r_best_flatfile	Flat Field File	R4?
w2r_best_maskfile	Static Mask File	R0?
w2r_best_shadfile	Shutter Shading Correction	R5?
w2r_best_comptab	Master Component Table	TMC.FITS
w2r_best_graphtab	The Master Graph Table (SYNPHOT)	TMG.FITS
w2r_best_idctab	Image Distortion Coefficients File	IDC.FITS
w2r_best_offtab	not used	-

Table A6: wfpc2_ref_data table reference useful keywords

Column_name	comment	type
w2r_obset_id	observation set id	-
w2r_obsnumA	observation number	base 36
w2r_atodgain	A-D Gain	electrons
w2r_equinox	equinox of celestial coord. system	-
w2r_expstart	Exposure start time	Modified Julian Date
w2r_filter1	First filter Numberf	-
w2r_filter2	Second filter Number	-
w2r_filtnam1	First filter Name	-
w2r_filtnam2	Secondfilter Name	-
w2r_mode	instrument mode	FULL (full res.), AREA (area int.)
w2r_orientat_1,2,3,4	Orientation of the image 1, 3, or 4	posangle
w2r_serials	serial clocks	ON, OFF
w2r_shutter	Shutter in place at beginning of the exposure	-
w2r_atodcorr	A-D correction applied	PERFORM, OMIT,COMPLETE
w2r_biascorr	Bias correction applied	PERFORM, OMIT,COMPLETE
w2r_blevcorr	Bias level correction applied	PERFORM, OMIT,COMPLETE
w2r_darkcorr	Dark correction applied	PERFORM, OMIT,COMPLETE
w2r_dophotom	Fill Photometry keywords	PERFORM, OMIT,COMPLETE
w2r_flatcorr	Flat correction applied	PERFORM, OMIT,COMPLETE
w2r_maskcorr	Mask correction applied	PERFORM, OMIT,COMPLETE
w2r_shadcorr	Shaded Shutter correction applied	PERFORM, OMIT,COMPLETE

NICMOS

Table A7: nicmos_ref_data table reference file identifier

Column_name	Reference file type	Reference file extension (type)
nsr_best_darkfile	Dark Current File	DRK
nsr_best_flatfile	Flat Field	FLT
nsr_best_illmfile	Illumniation Patern File	ILM
nsr_best_maskfile	On-Orbit MASK for NCS data	
nsr_best_nlinfile	Deterctor Linearity File	LIN
nsr_best_noisfile	Detector Read-Noise File	NOI
nsr_best_saadfile	Post SAA Dark	Assoc. Name
nsr_best_tempfile	temperature-dependent dark reference file	TDD
nsr_best_backtab	Background Model Table	-
nsr_best_phottab	Phototmetric Calibration Talbe	PHT
nsr_best_idctab	Image Distortion Coefficents File	IDC

Table A8: nicmos_ref_data table reference useful keywords

Column_name	comment	type
nsr_obset_id	Observation Set ID	-
nsr_camera	Camera in use	1, 2, or 3
nsr_expstart	Exposure Start Timei	MJD
nsr_filter	Filter Wheel Element	varchar
nsr_nread	Number of Initial and Final Readouts	small int
nsr_readout	Detector readout rate	FAST, SLOW
nsr_samp_seq	Number of Samples	int

Appendix B

From time to time we receive deliveries of new HST Calibration Spectra files or updates to the libraries of stellar models that SYNPHOT uses. Given that these files are used by all the instrument teams, there is no instrument in charge of the assessment and delivery of these files. Therefore, the INS/CDBS Team has taken the lead to support these activities, making sure that the files are CDBS compliant and tested by all the instrument teams. The bulk of the work on these files is on the testing. Although, it is not the duty of the INS/CDBS Team to do this work, it is important that the team makes sure that these files are tested with SYNPHOT (and if possible with the ETCs) and that the appropriate documentation describing the reason for the update appears in the header of the file.

Once the test has been completed by the instrument teams, the files can be copied to their respective directories in the CDBS area of the centralized storage. There is no need to deliver these files to the CDBS database; however, there are a series of steps that should be followed to make sure these files have the appropriate documentation and are in the appropriate directories.

First of all, there are two important steps that should always be followed when updating these files

1. The history of the new file should clearly state the reason for the update.
2. Since these files are used by all the instrument teams, they should sign-off the delivery of these files. We should request that instrument teams test these files; however, they might choose to sign-off without further testing. In any case, when they are aware of the changes and the possible implications the delivery of these files might have in the SYNPHOT or ETC calculations, it should be enough for us to make the delivery. Even in this case, a formal signoff, via e-mail should be required and kept in our records.

HST Standard Stars Spectra

In the case of HST Calibration Spectra, there are two independent directories containing the spectra of standard stars: CALOBS (/grp/hst/cdbs/calobs) and CALSPEC (/grp/hst/cdbs/calspec). CALOBS contains original as well as updated versions of the ultraviolet (IUE and VOYAGER2) and optical (Oke, Tapia or Stone) spectra of standard stars, while CALSPEC contains composite ultraviolet and optical absolute calibrated reference spectra of the HST standards.

Although these files are not delivered to the CDBS database, a set of several steps has to be followed when copying these to the centralized storage. The CALSPEC files have to be copied not only to the CALSPEC directory mentioned above but also to another directory called CURRENT CALSPEC (/grp/hst/cdbs/current_calspec/). This second directory can be found in the CDBS area in the centralized storage and has the same files that are in the CALSPEC directory. These two directories exist now to support SYNPHOT and ETC. In this case the directory CURRENT CALSPEC supports the ETCs while CALSPEC supports SYNPHOT. However, due to the current policies and procedures created by the HST Mission Office and the ETC Team for the ingest of all the SYNPHOT data files in the ETC servers, the CURRENT CALSPEC directory might become obsolete soon, except maybe for testing purposes. The need for this directory will have to be evaluated at a later time. In the mean time we still have to make the copy of these files in these two directories. Note that there is another directory called SUPPLEMENTAL CALSPEC (/grp/hst/cdbs/supplemental_calspec/) which contains a subset of HST calibration spectra. The files in these directories are different to those in the CALSPEC area and these are for calibration spectra of less accuracy than those in CALSPEC.

There are other things to consider about these files. The new files will follow the same version control as any other SYNPHOT file; that is, the version number increases by one for the new file being delivered. These are delivered by R. Bohlin, who is the person working on the HST standard spectra. He also helps us with the update of the CSBS Web page information for the CALSPEC files. Therefore, an update to the CALSPEC web page should be made every time a new HST Standard star is delivered.

SITS FUV-MAMA Dark Current Glow Image Files

The STIS FUV-MAMA dark current glow images files are in the CDBS area (/grp/hst/cdbs/stis_aux/) but are not delivered to the CDBS database. These files are auxiliary FUV MAMA dark files provided for users for five epochs between April 1997 and August 2004. For each of these epochs there is a dark mean dark rate including glow normalized to counts/pixel/second, a dark for hot pixels ($> 1e - 4c/s$) plus base level rate measured in dark corner and a dark with glow only (i.e. Mean dark minus hot pixels minus dark corner average). If STIS were to decide to update these files in the future, these can just be copied to the holding directory. We should, however, request that instrument team test these files against CALSTIS before delivery.

Grid of Spectral Atlases

CDBS stellar spectral atlases are a collection of theoretical models or composites of observational data that are well known and used by the astronomical community. These are used to model the spectra, to derive the photometry, or to determine the counts for a particular type of astronomical object. There is a considerable number of files of this kind, each grouped by the atlas they belong to. These files are rarely updated, so deliveries of this kind might not be seen in a long time. It is more likely that a new grid of models have to be added than that the models need to be changed. In the case a new grid needs to be added, a detailed description of the models have to be included in these directories as a readme file together with the set of models. This README file has to be added to the CDBS web site too. Currently we have 12 atlases in the CDBS GRID directory:

1. CASTELLI-KURUCZ ATLAS. It contains about 4300 stellar atmosphere models.
2. PICKLES ATLAS. This library of wide spectral coverage, consists of 131 flux calibrated stellar spectra.

3. BUSER-KURUCZ ATLAS. The catalog consists of 1434 files.
4. KURUCZ 1993 ATLAS contains about 7600 stellar atmosphere models.
5. BRUZUAL ATLAS contains 77 stellar spectra.
6. GUNN-STRYKER ATLAS consists of 175 spectra of stars.
7. BRUZUAL-PERSSON-GUNN-STRYKER ATLAS contains 175 spectra.
8. JACOBY-HUNTER-CHRISTIAN ATLAS contains 161 spectra of stars.
9. BRUZUAL-CHARLOT ATLAS is a library of 84 galaxy spectra.
10. KINNEY-CALZETTI ATLAS consists of an homogeneous set of 12 spectral templates of galaxies
11. AGN ATLAS consists of 5 spectral templates of AGNs ranging from LINER to Seyfert and bright QSO.
12. GALACTIC ATLAS consists of model spectra of the Orion Nebula and of the NGC 7009 planetary Nebula.