Technical Report MESA 2021-01

Preparation and Delivery of Calibration Reference Files

Matt McMaster[1]

[1]*Space Telescope Science Institute, Baltimore, MD*

November 11, 2021

ABSTRACT

This document gives information regarding what instrument teams need to do prior to the submission of calibration reference files to CRDS via the Extended Batch Submit web form and the ReDCaT Team, as well as discusses important aspects of the CRDS systems, management of reference files and rmaps that is not discussed elsewhere. Also included is information regarding what to do when a new reference file type is needed, or when making a significant change to an existing reference file type (new modes, different number of extensions, new columns in tables, etc.). Where appropriate, the information will be split according to telescope, since the procedures can vary slightly for each. Instructions on delivering reference files using the Extended Batch Submit webform are provided via links in Section 6. Finally, a section giving an overview of the various mapping files.

1 Introduction

The generation of calibration reference files is the responsibility of the instrument team whose data will be calibrated by them. As part of this process, testing and assessment of the reference files needs to be performed before they are delivered to DMS (Data Management System), which includes CRDS (Calibration Reference Data System) and DADS (Data Archive Distribution Services), or the archive. Ingestion into the archive allows both local and offsite users to retrieve them. Note that users can also obtain references using CRDS tools. Information on these tools can be found at: https://hst-crds.stsci.edu/static/users_guide/basic_use.html or https://jwst-crds.stsci.edu/static/users_guide/basic_use.html.

Along with certifying reference files and having them ingested into the archive and various caches, CRDS also selects references for a given observation mode of the data. The selection of references is based on information contained in CRDS rmaps (rules or reference mapping

files – see Section 8.3). For the HST instruments, the selection rules are based on what is contained in ICD-47 (Interface Control Document), located at: http://newcdbs.stsci.edu/doc/ICD47/Section1.html. For JWST instruments, the selection rules are generated from information given in the calibration reference files document, located at: https://jwst-pipeline.readthedocs.io/en/latest/jwst/introduction.html#reference-files.

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

- ensures that the required keyword content and syntax are correct
- ensures that file structure and format are correct
- ensures that the reference files work properly with the calibration software
- ensures that the reference files are properly selected by CRDS
- verifies scientific validity
- ensures proper documentation of the reference files (though for JWST, the responsibility here lies with the Data Management System Working Group, since the files are, for the most part, uniform across instruments)

The instrument branches are responsible for all of the points listed above, while during delivery, CRDS performs checks to verify compliance of the first four points. Note that reference files are the responsibility of the instrument branches and that no changes to the files, regardless of how seemingly insignificant they may be, will be made by the ReDCaT Team.

2 Information on File Preparation

HST reference files are all in FITS format, while JWST references can be FITS, ASDF, or JSON files. Regardless of the format, all reference files must be compliant to the standards for that format type. If you are using python when generating a file, you will receive a warning or error upon closing the file if it is not compliant. If this happens, the file must be corrected or it may be rejected by CRDS. Along with having your references format compliant, there are several keywords that are also required. There are keywords that are mission specific, and keywords that are reference file specific.

2.1 HST

For HST, the mission specific keywords are: USEAFTER, COMMENT, DESCRIP, PEDIGREE, and HISTORY.

2.1.1 USEAFTER

The USEAFTER keyword has the format, MMM DD YYYY HH:MM:SS, and is the date and time (UT) after which the reference file should be used; all of the fields are required. If there is no meaningful time for a reference file (e.g., a reference file that covers data throughout the life of the instrument), it should be set to 00:00:00. Note, if the reference file being delivered replaces one currently in use, the **USEAFTERs have to match exactly** (to the second); not doing so will result in either an orphaned reference file (an active reference file that will not be used on any data), or incomplete coverage (some data will not be calibrated with the new reference file). If the reference being replaced should no longer be used for any reason, note this when submitting your files.

2.1.2 COMMENT

The COMMENT keyword is used to designate who created the reference file, and can include any script used in generating it (e.g., "ACS WFC bias created by M. McDonald", or "Reference file created by B. Kuhn using scriptname.py"). Note that some FITS files contain the default COMMENT section which gives the literary reference about these types of files; this section should be removed before your reference files are submitted (e.g., entries like the following section should not be in your files:

```
COMMENT   FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT   Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT   Contact the NASA Science Office of Standards and Technology for the
COMMENT   FITS Definition document #100 and other FITS information).
```

2.1.3 DESCRIP

The DESCRIP keyword should be a concise sentence that makes it easy for a user to determine the type of reference file and the changes being made; it should be a one-line explanation of why the file is being delivered. For example:

DESCRIP = "ACS cte corrected dark for data taken after May 21 2016."
DESCRIP = "New STIS MAMA blaze function corrections and sensitivities for echelle modes."

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

2.1.4 PEDIGREE

The PEDIGREE keyword gives the source of the data used in generating the reference file, and can have the value of DUMMY, MODEL, GROUND, or INFLIGHT

2.1.4.1 DUMMY

A placeholder file which often contains only ones or zeroes, depending on whether the reference file is multiplicative or additive, respectively.

2.1.4.2 MODEL

A reference file generated using only model or mathematical information.

2.1.4.3 GROUND

A reference file created using data obtained during ground testing.

2.1.4.4 INFLIGHT

A reference file created by utilizing data taken on-orbit. It has the format:
INFLIGHT DD/MM/YYYY DD/MM/YYYY
which represents the range of dates of the data used in generating the file.

2.1.5 HISTORY

The HISTORY section should contain a complete, but concise description of the process used in generating the reference file, including any scripts used, the total exposure time, the timeframe over which the exposures used in creating the reference were taken, the value above which hot pixels were flagged, etc. Other important information would be the names of the datasets used in generating the file, pointers to relevant documentation, why the file is being delivered and/or an explanation of what was updated, etc. As much as possible, the

information should pertain to just that particular file and not the reference type as a whole, though pointers to such documentation are fine (e.g., “For more information on how WFC3 darks are generated, see WFC3 ISR 2018-08.”).

2.2 JWST

The JWST mission specific keywords are: REFTYPE, DESCRIP, AUTHOR, USEAFTER, PEDIGREE, and HISTORY.

2.2.1 REFTYPE

The type of reference file being delivered. The acceptable values are listed in
https://jwst-pipeline.readthedocs.io/en/stable/jwst/references_general/references_general.html#reference-file-types

2.2.2 DESCRIP

This keyword gives a summary of the file content and the reason for delivery; it should be a concise sentence. For example:

DESCRIP = “New NIRISS pixel flat for the F356W filter updated with cycle 1 observations.”
DESCRIP = “New MIRI MIRIMAGE readnoise array updated with cycle 2 observations.”

2.2.3 AUTHOR

The person or persons who created the file. For example:

AUTHOR = “M. Cracraft”
AUTHOR = “B. Hilbert and A. Canipe”

2.2.4 USEAFTER

The USEAFTER keyword has the format YYYY-MM-DDTHH:MM:SS, which is the date and time (UT) after which the reference file is to be used; the T is required. If there is no meaningful value for the time, it should be set to 00:00:00.

2.2.5 PEDIGREE

As is the case for HST reference files, PEDIGREE gives the source of the data used in generating the reference file. The acceptable values are: MODEL, GROUND, DUMMY, or INFLIGHT.

2.2.5.1 MODEL

A reference file created using modeling or simulation software.

2.2.5.2 GROUND

GROUND is used for a reference file derived from data taken during ground testing.

2.2.5.3 DUMMY

A placeholder reference file which may contain all zeroes (additive) or ones (multiplicative).

2.2.5.4 INFLIGHT

Use INFLIGHT for a reference file derived from data taken on-orbit, having the format: INFLIGHT YYYY-MM-DD YYYY-MM-DD, which represents the range of dates of when the data used to generate the reference file were taken.

2.2.6 HISTORY

The HISTORY section contains a detailed description of the creation of the reference file, including documentation which describes the strategy and algorithms used to generate the file, software used, the names of the raw data files used, the differences with the file it is replacing (if applicable). As much as possible, the information contained in the HISTORY section should pertain to that particular reference and not the reference type as a whole, though pointers to such information are fine. The HISTORY section should look something like the following:

HISTORY Date corresponding to the following description
HISTORY Description of how the reference was created.
HISTORY DOCUMENT: Title of the document describing the strategy and algorithms used
HISTORY in generating the file
HISTORY SOFTWARE: The description, version number, and location of the software that
HISTORY was used in creating the file.
HISTORY DATA USED: Names of the data files used to create the file.
HISTORY DIFFERENCES: If applicable, how is this version of the reference file different
HISTORY different than the one being replaced.

2.3 Other Important Keywords

There are other keywords, specific to instrument and reference type, that are also required. There are a couple of ways in which to find them. One is to examine the rmap (details about rmaps can be found in Section 8.3). For HST references, go to https://hst-crds.stsci.edu; similarly, for JWST instruments, go to https://jwst-crds.stsci.edu. Click on your instrument and then on the reference type. This is shown in Figure 2-1.

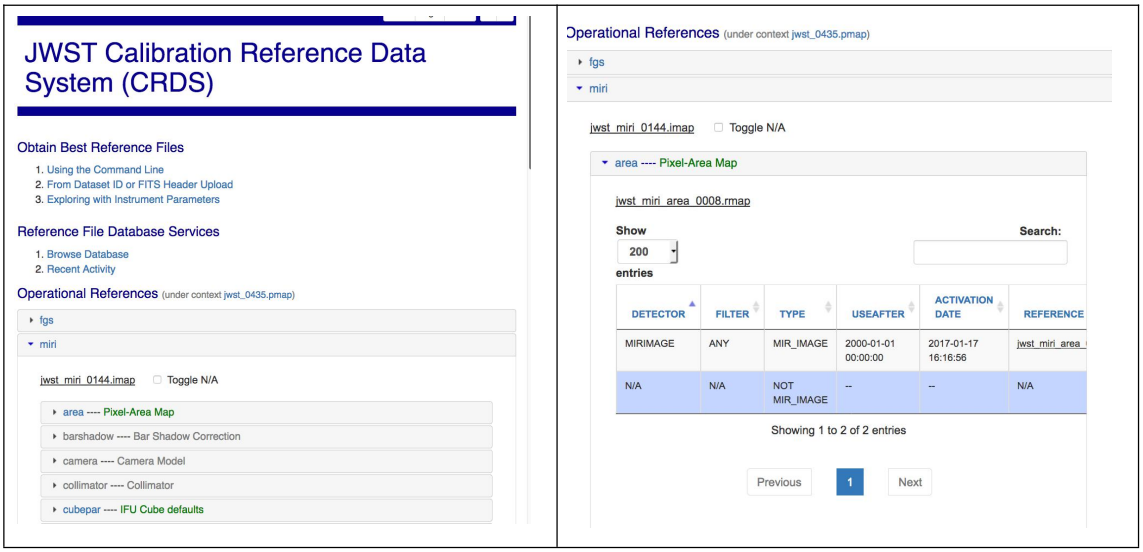


Figure 2-1 Sample view from jwst-crds.stsci.edu

At the top of the listing, you will see a link to the rmap (e.g., jwst_miri_area_0008.rmap). Clicking on that link will take you to another page with four dropdowns; select (open) the one marked Contents. Partway down, you will see an entry called 'parkey', short for parameter keyword. See Figure 2-2 below.

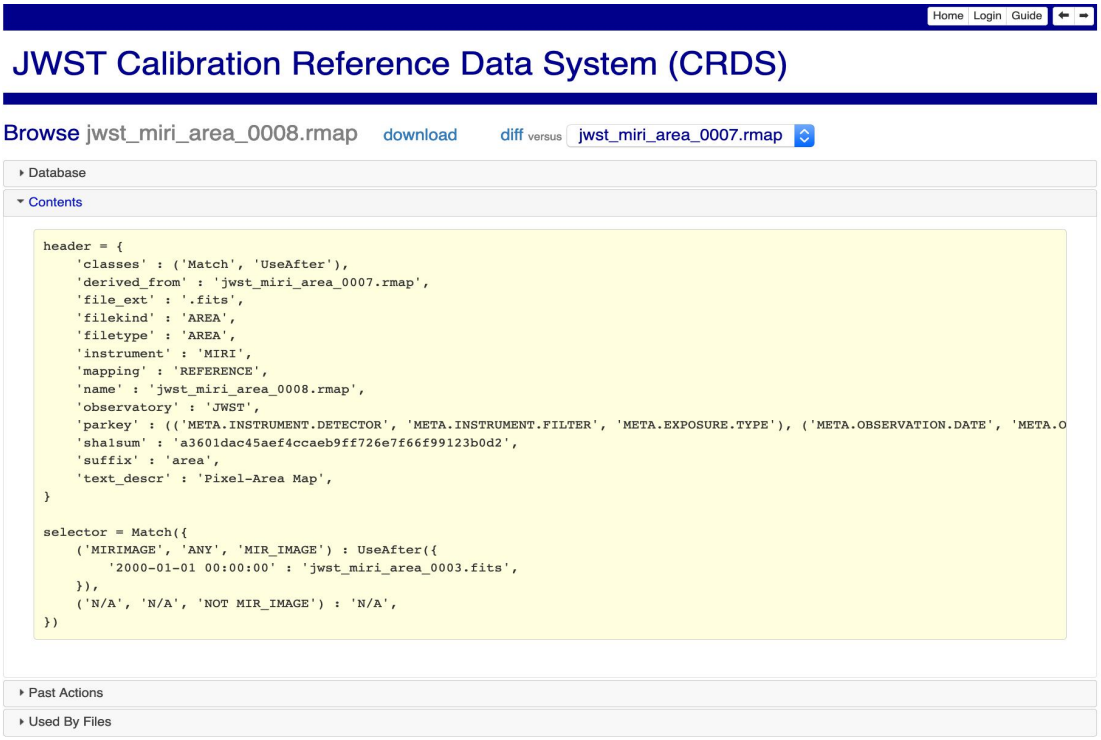


Figure 2-2 Sample view for rmap files

The view above does not show the full `parkey` entry. In this case it is:

```
parkey' : (('META.INSTRUMENT.DETECTOR', 'META.INSTRUMENT.FILTER',
'META.EXPOSURE.TYPE'), ('META.OBSERVATION.DATE', 'META.OBSERVATION.TIME')),
```

These are the keywords that CRDS pulls from the data in order to determine the best reference file to use, and are also the required keywords for that particular reference file type. For JWST, it's the meta keywords that are listed (e.g., META.SUBARRAY.NAME, META.INSTRUMENT.DETECTOR) but their header keyword counterparts (SUBARRAY, DETECTOR, respectively) are the ones that are required in the reference file. Note that in the `selector = Match` section, DATE-OBS and TIME-OBS are combined to generate a date which is used for comparison with the USEAFTERs of the reference files.

Another way to determine other needed keywords is to consult the reference file documentation. For HST, this is the ICD47 (Instrument Control Document at: http://newcdbs.stsci.edu/doc/ICD47/Section1.html). Go to the appropriate instrument section, find the given reference file type, and the information is contained in the Selection Criteria and/or Required Additional Keywords sections. Information on JWST reference files can be found on the Reference File Information page, in the Reference File Types section: (https://jwst-pipeline.readthedocs.io/en/stable/jwst/references_general/references_general.html#reference-file-types).

3 FILE NAMES

There are no restrictions on what you name your reference files when they are submitted to CRDS for delivery, the files will be renamed based on their FILETYPE (HST) or REFTYPE (JWST) keywords.

3.1 HST

HST reference files are renamed by CRDS during the delivery process via crds.uniqname. A reference file is named based on the years since HST launch, date and time (UT), instrument, and reference type, and has the form:

```
ymdhhmmsi_type.fits
```

where

y = year (a-z, 0-9, reference files delivered in 2021 all begin with 5)
m = month (1-9, a-c)
d = day (1-9, a-s, [t], [u], [v])
hh = UT hours (00-23)
mm = UT minutes (00-59)
s = UT seconds, in 2 second intervals (0-9, a-t)
i = instrument (j = ACS, l = COS, o = STIS, i = WFC3)
type = reference type extension (e.g., bia, drk, gsag).

3.2 JWST

For JWST reference files, the names are generated with sequentially increasing version numbers by CRDS during the file certification process, and have the form:

```
jwst_instrument_reftype_version.ext
```

where

instrument = fgs, nircam, niriss, nirspec, or miri
reftype = the name of the reference file type (e.g., superbias, dark, area)
version = a four digit version number starting at 0001
ext = fits, asdf, json

4 CERTIFYING FILES

While CRDS will certify your reference files when they're submitted, it's a good idea to pre-certify them in order to address any issues that might arise and cause unnecessary delays in the delivery. This can be done via command line, or one of the CRDS web pages (https://hst-crds.stsci.edu or https://jwst-crds.stsci.edu), provided you have the ability to deliver reference files using the Extended Batch Submit webform (see below).

For the command line version, start a bash session and then activate your usual conda environment, making sure to update both:

```
mycomputer: bash
mycomputer: conda update --yes -n base -c defaults conda
(base) bash$ conda activate astroconda (or whatever your usual conda
environment is called)
(astroconda) bash$ conda update --yes --all

(astroconda) bash$ crds certify /full/path/to/your/files/*.fits (or *.json,
or *.asdf)
```

or, if you're in the directory where your files reside:

```
(astroconda) bash$ crds certify ./*.fits (or *.json, or *.asdf).
```

Note that along with ensuring that your files are CRDS compliant, crds certify also ensures that your files are format compliant (fits, json, or asdf).

5 TESTING FILES

During delivery, your files will be checked for fits (or asdf, or json) compliance, required keywords are present and contain valid values, and that the generated matching rules don't interfere with proper reference file selection. However, the scientific validity of the files are not checked (neither the ReDCaT Team nor CRDS check to make sure that using the reference file in calibration produces expected results). Therefore, it is highly recommended that reference files be thoroughly checked before submission. It is also recommended that you verify that the new reference files have the desired effect on the data they'll be used on. Note how you determined that the files were good, since this information is needed during submission.

To ensure that there are no problems within the calibration pipeline, you are also required to test the reference using the version of the calibration software currently being used in operations. For JWST references, you can directly use the Calibration Pipeline step you need.

If your reference files are first going into the TEST system in order to test a new, unreleased version of the calibration software, this must be noted when the files are submitted. Note, this is particularly important when a reference file is new or has a format change which depends on the software used. We need to test that the software and files work well together before they can be submitted to the operational system (OPS).

6 DELIVERING FILES

Previously, reference files were delivered by the ReDCaT Team using information provided on a delivery form submitted by an instrument branch. Now, however, certain members of an instrument branch, who have been given CRDS access, can deliver reference files by means of the Extended Batch Submit webform, located in the Routine Submission Services area (after signing in) on the top page of any CRDS server. See Figure 6-1 below for an example.

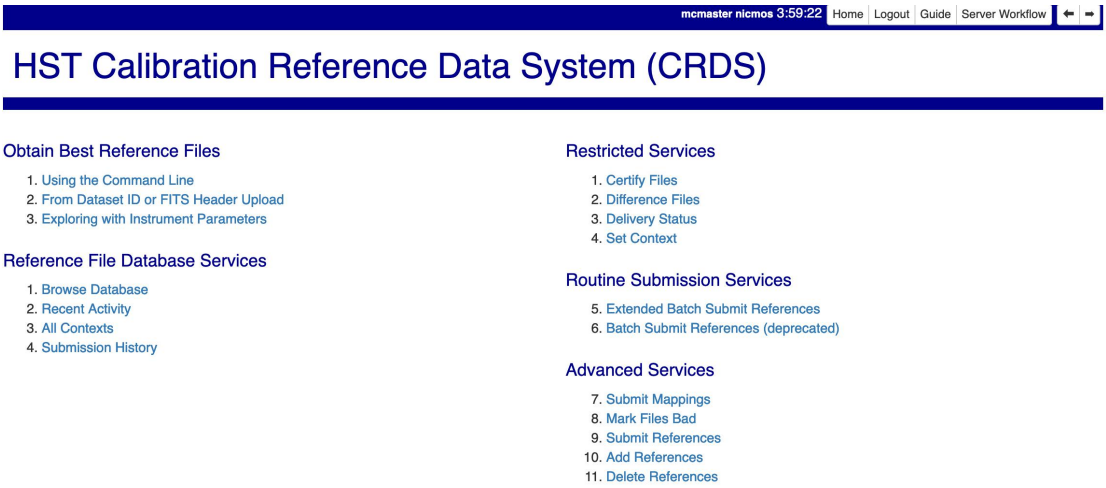


Figure 6-1 CRDS Front Page with Extended Batch Submit Link (item 5)

6.1 Extended Batch Submission Instructions

Before you can make deliveries, you need to have access to the CRDS system. To do so, contact the ReDCaT Team representative for your instrument, or send a request to Matt McMaster (mcmaster@stsci.edu). Instrument branches are requested to keep the number of submitters to a minimum in order to limit the possibility of a collision during deliveries, as well as for security purposes.

Prior to submitting your reference files, inform the ReDCaT Team on the redcat_and_friends slack channel that you will be making a submission. If there is anything going on where deliveries can't occur (server down, the instrument section you need is being used by someone else, etc.), you will be asked to hold your delivery, and you will be notified when you can proceed.

Follow the instructions on signing on to the server, uploading your files, and submitting them to CRDS given on our Confluence page at:
https://innerspace.stsci.edu/display/RT/Delivery+of+Operational+Reference+Files

7 DEFINING A NEW REFERENCE FILE TYPE

Occasionally, a new reference file type may be required by an instrument branch, either due to a change in the calibration software, or as a mission requirement. In order for CRDS to properly handle the new type, it needs to be updated to properly certify and ingest the file. To do this, file a JIRA issue against CRDS (https://jira.stsci.edu/browse/CRDS), and assign it to Matt McMaster (or the ReDCaT Team Lead). Fill in the following information, and add it as the description in your issue:

```
Basic Information
=================
1- JIRA ISSUE number
-------------------------------------------
2- FILE DESCRIPTION (For rmap and web submission documentation)
-------------------------------------------------------------
3- INSTRUMENT
-------------------
4- TYPE NAME (CRDS filekind, no underscores, letter first; e.g.ipc, mask)
-------------------------------------------------------------------------
5- LONG FORM TYPENAME (CRDS filetype? for website context display;e.g.
“Interpixel Capacitance”, “Bad Pixel Mask”)
-------------------------------------------------------------------------
6- EXTENSION / FILE FORMAT (.asdf, .fits, .json)
-----------------------------------------------------------
7- SELECTION PARAMETERS (e.g. DETECTOR, FILTER)
-----------------------------------------------------------------------
8 - REQUIRED (YES / NO?)
-------------------------------
9- DIRECTORY PATH WHERE WE CAN FIND THE FILE(S)
-------------------------------------------------------------------------
10- INTIAL REFERENCE FILENAMES (e.g. jwst_fgs_ipc)
-----------------------------------------------------------------
11- FILE CREATOR
----------------------
12- TABLE (YES / NO?)
-------------------------
13- PARAMETER NAME FOR CALIBRATION STEP (eg. HST BIASCORR)
-------------------------------------------------------------------------
14- OBSERVATORY
----------------------
15- WHEN NOT USED (N/A):
--------------------------------
16- New REFTYPE, EXP_TYPE, or other new selection criteria (JWST only)

More Advanced
=============
17- EXPRESSION DEFINING N/A (rmap_relevance expression e.g.DEADCORR!="OMIT",
OPTIONAL)
-------------------------------------------------------------------------
18- PARAMETERS DEFINING N/A (cases with N/A as filename due to parameter
value combinations which should be defined as having N/A results, OPTIONAL)
-------------------------------------------------------------------------
19- PARAMETER VALUE SUBSTITUTIONS (e.g. GENERIC --> N/A, reference +
displays as GENERIC, CRDS matches as N/A, OPTIONAL)
-------------------------------------------------------------------------
20- CONDITIONAL PARAMETER VALUE SUBSTITUTIONS (e.g. WFC3 APERTURE group
translations, rmap update time, "LRFWAVE == '3774.0'" : 'between 3710 3826’,
"DETECTOR=='HRC' and CCDGAIN=='-1'": '1.0|2.0|4.0|8.0’, OPTIONAL)
-------------------------------------------------------------------------
21- UNIQUE ROW COLUMN NAMES (for table difference checking, OPTIONAL, tables)
-------------------------------------------------------------------------
22- TABLE ROW LOOKUP COLUMN NAMES (for reprocessing optimization,
OPTIONAL, tables)
-------------------------------------------------------------------------
23- TABLE ROW LOOKUP ALGORITHM (for reprocessing optimization, OPTIONAL,
tables)
-------------------------------------------------------------------------
```

Items:

1- JIRA Issue: The JIRA ISSUE that is being filed for the new reference type, plus any other relevant issues (e.g., CRDS-122). You might not know this when you are filing the JIRA issue, so you can add it after submission.

2- FILE DESCRIPTION: A short description of the new file type for the rmap and web submission documentation. For example, NIRSpec IRS2 Reference Pixel File

3- INSTRUMENT: Name of the instrument.

4- TYPE NAME: The reference file type; this will be the REFTYPE keyword for JWST reference files and the FILETYPE keyword for HST. It will also be used as the 'filekind ' entry in the rmap. Examples of this are: 'dark 'or 'mask 'for the Dark Frame or the Bad Pixel Mask, respectively.

5- LONG FORM TYPENAME: For example," Interpixel Capacitance" for the *_ipc_*.fits files or "Bad Pixel Mask" for *_mask_*.fits files.

6- EXTENSION / FILE FORMAT: The file format of the reference. CRDS can handle FITS, JSON, ASDF, and YAML formats see:
https://jwst-crds.stsci.edu/static/users_guide/reference_conventions.html or
https://hst-crds.stsci.edu/static/users_guide/reference_conventions.html

7- SELECTION PARAMETERS: The list of keywords you want the reference file to be selected on; e.g., DETECTOR, EXP_TYPE, FILTER, PUPIL.

8- REQUIRED: This is almost always YES. What's being asked is: if a reference file can't be found based on the values of the keywords used in the selection parameters provided in item 7, do you want CRDS to issue an error and stop any further processing? If YES, then CRDS will issue an error and exit from any further processing of the data. If NO, then there won't be an error issued and processing will continue.

9- DIRECTORY PATH: The directory where samples of the new reference type can be found.

10- INTIAL REFERENCE FILENAMES: The names of the files in the directory above that can be used to test CRDS. These do not need to be the references that will be delivered, but their headers, etc., need to be set up to ensure that they will be accepted by CRDS.

11- FILE CREATOR: The person who generated the test reference files above, or someone who can be contacted in case of questions.

12- TABLE (YES / NO?): Is the new reference a table?

13- PARAMETER NAME FOR CALIBRATION STEP (e.g., HST BIASCORR): The keyword which will be used to trigger the calibration step where this reference file will be used (e.g., BIASCORR, SINKCORR). This is for HST only.

14- OBSERVATORY: HST, JWST, or NGRST/Roman

15- WHEN NOT USED (N/A): If the new reference type makes another reference type obsolete, list the obsolete reference. For example, when the COS Branch introduced their XWLKFILE and YWLKFILE reference files, the WALKTAB reference file was no longer

needed and was removed from the COS imap (see Section 8.2 for more information on the imap files). In this case, WALKTAB would have been entered.

16- New REFTYPE, EXP_TYPE, or other new selection criteria (JWST only): The fundamental drivers for the CRDS reprocessing code are EXP_TYPE and the calibration pipeline stage, in some cases, other parameters might also be important. If there are new observing modes being handled, because of a new EXP_TYPE or other selection criteria, CRDS reprocessing needs to know which calibration pipelines are expected to be used for routine calibration, and possible exceptions. This enables CRDS reprocessing to determine which CAL steps run, and from the steps, which reference file types should be checked for potential reprocessing effects. Specify which pipeline(s), either by alias name, class name, or parameter reference file name, are expected to be affected by this new reference file type and modes it is covering. If you are unsure what to put, answer: Unknown.

For more information about the calibration steps, refer to 'Stages of Processing 'at:
https://jwst-docs.stsci.edu/jwst-science-calibration-pipeline-overview/stages-of-jwst-data-processing

17- EXPRESSION DEFINING OMISSION (rmap_relevance expression e.g., DEADCORR!= "OMIT", OPTIONAL): This is for HST reference files, and is a single expression that defines when a reference file type should be omitted, and is generally used by HST to incorporate logic related to CORR keywords that was not included in the old CDBS reference file templates. There can be only one expression per rmap, and it should be relatively simple. Examples:

- DETECTOR == WFC and PCTECORR != OMIT (the DETECTOR has to be WFC and PCTECORR cannot be OMIT in order for the reference file to be used).
- DETECTOR == FUV (the reference file can only be used for observations taken in the FUV detector).
- LIFE_ADJ in ["3.0", "4.0"] (the LIFE_ADJ keyword has to be either 3.0 or 4.0 in order for the reference file to be used).

18- PARAMETERS DEFINING OMISSION (OPTIONAL): This is for JWST reference files, and is where a match case is not applicable (N/A) for that particular calibration step, and hence should not return a specific reference file. It can be as simple or complicated as need be, there can be more than one, and are specified using match case notations. They can also be seen in the operational context display on https://jwst-crds.stsci.edu

For example, for the NIRSpec wavelengthrange reference type, the file is selected on the value of EXP_TYPE and the date and time of the observation. However, this reference file is not applied to those data types that are darks (NRS_DARK). This case is denoted within the rmap as:

- 'NRS_DARK' : 'N/A'

which tells CRDS that this reference file type is omitted for this type of data.

19- RMAP PARAMETER VALUE SUBSTITUTIONS: For a given keyword used in selecting the reference file, if CRDS sees this value, what should it match as. For example, one of the selection criteria for JWST superbias reference files is the value of the SUBARRAY keyword. For NIRSpec, if the value of the keyword in the superbias reference is GENERIC, that file will be used for the cases where the SUBARRAY of the data can be ignored. The entry in the rmap for this looks like:

```
'substitutions' : {
    'META.SUBARRAY.NAME' : {
        'GENERIC' : 'N/A',
```

```
    },
},
```

This is known as an unconditional substitution because any instance of it has exactly that translation, regardless of any other parameter values. This feature has the advantage that it can be used without changes to CRDS code, modifying only the rmaps. It is limited to translating simple string values, so is not usable in all cases. Note that defining substitutions may require coordination with the CAL code data models and changes to schema.

20- CONDITIONAL PARAMETER VALUE SUBSTITUTIONS: The set of conditions one or more header keywords must meet in order for the reference file to be applied. For example:

```
"DETECTOR=='NRCA1' and EXP_TYPE=='NRC_IMAGE'"
'((DETECTOR == "FUV") and (OBSMODE=="TIME-TAG") and (XWLKCORR != "OMIT"))'
```

21- UNIQUE ROW COLUMN NAMES (for table difference checking, OPTIONAL, tables): If the reference file you're defining is a table, list the column names you'd like CRDS to use in its comparison to the previous version of the reference (note, you do not need to list all of the column names). CRDS attempts to check for duplicate rows or rows accidentally deleted from a new version of a table. These column names are used to give rows an identity that can be used to discern if there is more than one instance, or if some combination that was present in the prior table is not present in the new version. Not all columns are used to define identity so that, for example, coefficients can be tweaked between versions of a table without making it appear that something has been deleted. This also is information that will be used to ensure that the reference table has all of the expected value combinations and checking for completeness and duplication.

22- TABLE ROW LOOKUP COLUMN NAMES (for reprocessing optimization, OPTIONAL, tables): The name(s) of the column(s) that are most likely to change and would warrant the reprocessing of affected data. This information will be used to determine what data need to be reprocessed, as well as avoid unnecessary reprocessing of data that was not affected by the delivery of the reference file.

23- TABLE ROW LOOKUP ALGORITHM (for reprocessing optimization, OPTIONAL, tables): The table row lookup algorithm can be used by CRDS reprocessing to determine which rows in a table affect a particular dataset parameter combination. CRDS will use the algorithm and dataset parameters to attempt to select the same table rows that would be used in calibration. This enables CRDS to determine if a table update affects a particular dataset based on the rows that were changed or if reprocessing is not needed (this is currently unused).

Once the above form is complete, attach it to your JIRA issue. While the issue is being worked, you may be contacted for details regarding the new reference file type. Be ready to provide samples of the files (location is included in the form). These do not have to be the actual files you plan to deliver, just something that can be used to ensure that the references will be accepted by CRDS. Finally, wait until you are contacted by the ReDCaT Team before delivering your files.

8 MAPPING FILES

This section is a brief overview of the mapping files used by CRDS. For more in depth information, see the CRDS rules section of the CRDS users guide at: https://hst-crds.stsci.edu/static/users_guide/overview.html#crds-rules or https://jwst-crds.stsci.edu/static/users_guide/overview.html#crds-rules

CRDS uses a series of mapping files to 'drill down' in order to find the best reference files for a given dataset, using a set of rules provided by the instrument branches. The progression goes from the pmap, also known as the context (or pipeline or processing map), to the imap (or instrument map), to the rmap (rules/reference map).

8.1 The PMAP

The pmap contains the list of imaps associated with it, and provides a link to all of the references files that are used with it. There can be only one PMAP in use within the calibration pipeline, but you can point to any existing pmap using one of the methods below (note that you must be in a bash shell using the latest version of astroconda). Note that, the addition (or removal) of a single file will result in the creation of a new pmap. Users can determine the reference files that were in use when their data was calibrated by identifying the PMAP that was operational at that time.

For more information on how to point to a given pmap, and update your headers to use the reference files for it, see the Environment Variables (https://hst-crds.stsci.edu/static/users_guide/environment.html) and Best Reference Basics (https://jwst-crds.stsci.edu/static/users_guide/basic_use.html) sections in the CRDS User Guide.

Below are examples for both the HST (Figure 8-1) and JWST (Figure 8-2) pmaps.

```
header = {
    'derived_from' : 'hst_0846.pmap',
    'mapping' : 'PIPELINE',
    'name' : 'hst_0847.pmap',
    'observatory' : 'HST',
    'parkey' : ('INSTRUME',),
    'sha1sum' : '558c03ab8633ac6bb5f457e2a7be2edc8a29a0b8',
}

selector = {
    'ACS' : 'hst_acs_0452.imap',
    'COS' : 'hst_cos_0320.imap',
    'NICMOS' : 'hst_nicmos_0250.imap',
    'STIS' : 'hst_stis_0321.imap',
    'SYNPHOT' : 'hst_synphot_0035.imap',
    'WFC3' : 'hst_wfc3_0468.imap',
    'WFPC2' : 'hst_wfpc2_0250.imap',
}
```

Figure 8-1: HST pmap

```
header = {
    'derived_from' : 'jwst_0643.pmap',
    'description' : 'Hand-edit to define all step parameter reftypes for all instruments',
    'mapping' : 'PIPELINE',
    'name' : 'jwst_0644.pmap',
    'observatory' : 'JWST',
    'parkey' : ('META.INSTRUMENT.NAME',),
    'sha1sum' : '5f079ad90042add7936ba5b6dfd5dbf48c4c2369',
}

selector = {
    'FGS' : 'jwst_fgs_0083.imap',
    'MIRI' : 'jwst_miri_0207.imap',
    'NIRCAM' : 'jwst_nircam_0144.imap',
    'NIRISS' : 'jwst_niriss_0136.imap',
    'NIRSPEC' : 'jwst_nirspec_0211.imap',
    'SYSTEM' : 'jwst_system_0022.imap',
}
```

Figure 8-2: JWST pmap

8.2 The IMAP

The imap (instrument map) files contain a list of the rmaps for that particular instrument. For JWST, the imaps contain all reference file types, but only have rmaps listed for those types actually being used to calibrate data from that instrument. If a reference type is not used or needed for a particular calibration pipeline step, it will be listed as 'N/A' in the imap. For example, the BARSHADOW reference type rmap is listed as 'N/A' for MIRI. The imaps for the HST instruments contain only those rmaps relevant to that instrument (i.e., none of the listings are set to 'N/A').

In both of the image above, the 'parkey' value, INSTRUME for HST and META.INSTRUMENT.NAME for JWST, given as the INSTRUME keyword in the data header, determines which imap CRDS will use.

Examples of the ACS (Figure 8-3) and MIRI (Figure 8-4) imaps are shown below. Notice that none of the reference types for ACS are listed as ‘N/A’, while a number are listed for MIRI (e.g., CAMERA, REFPIX, SUPERBIAS).

```
header = {
    'derived_from' : 'hst_acs_0452.imap',
    'instrument' : 'ACS',
    'mapping' : 'INSTRUMENT',
    'name' : 'hst_acs_0453.imap',
    'observatory' : 'HST',
    'parkey' : ('REFTYPE',),
    'sha1sum' : '7b2fef4d6235dd158c54ae4403208944b1475547',
}

selector = {
    'atodtab' : 'hst_acs_atodtab_0251.rmap',
    'biasfile' : 'hst_acs_biasfile_0362.rmap',
    'bpixtab' : 'hst_acs_bpixtab_0252.rmap',
    'ccdtab' : 'hst_acs_ccdtab_0255.rmap',
    'cfltfile' : 'hst_acs_cfltfile_0250.rmap',
    'crrejtab' : 'hst_acs_crrejtab_0251.rmap',
    'd2imfile' : 'hst_acs_d2imfile_0253.rmap',
    'darkfile' : 'hst_acs_darkfile_0383.rmap',
    'dgeofile' : 'hst_acs_dgeofile_0250.rmap',
    'drkcfile' : 'hst_acs_drkcfile_0394.rmap',
    'flshfile' : 'hst_acs_flshfile_0263.rmap',
    'idctab' : 'hst_acs_idctab_0256.rmap',
    'imphttab' : 'hst_acs_imphttab_0256.rmap',
    'mdriztab' : 'hst_acs_mdriztab_0253.rmap',
    'mlintab' : 'hst_acs_mlintab_0250.rmap',
    'npolfile' : 'hst_acs_npolfile_0253.rmap',
    'oscntab' : 'hst_acs_oscntab_0251.rmap',
    'pctetab' : 'hst_acs_pctetab_0253.rmap',
    'pfltfile' : 'hst_acs_pfltfile_0253.rmap',
    'shadfile' : 'hst_acs_shadfile_0251.rmap',
    'snkcfile' : 'hst_acs_snkcfile_0046.rmap',
    'spottab' : 'hst_acs_spottab_0250.rmap',
}
```

Figure 8-3: ACS imap

For HST data, if a calibration step is to be done (####CORR set to PERFORM) CRDS will check the imap for the appropriate rmap, and will then select the best reference file based on the rules it has. In the case of JWST data, CRDS will examine the imap as it progresses through the various calibration steps. If an rmap is listed, it will be opened and an appropriate reference file will be chosen based on the selection rules CRDS has. However, if a reference type in the imap is listed as N/A, CRDS will not select a file since the calibration step that would use that reference type will be skipped for that instrument.

```
header = {
    'derived_from' : 'jwst_miri_0205.imap',
    'instrument' : 'MIRI',
    'mapping' : 'INSTRUMENT',
    'name' : 'jwst_miri_0206.imap',
    'observatory' : 'JWST',
    'parkey' : ('REFTYPE',),
    'sha1sum' : '15b4f14d49e2e49a9f272bad2517776ac438f50d',
}

selector = {
    'ABVEGAOFFSET' : 'jwst_miri_abvegaoffset_0002.rmap',
    'APCORR' : 'jwst_miri_apcorr_0005.rmap',
    'AREA' : 'jwst_miri_area_0011.rmap',
    'BARSHADOW' : 'N/A',
    'CAMERA' : 'N/A',
    'COLLIMATOR' : 'N/A',
    'CUBEPAR' : 'jwst_miri_cubepar_0006.rmap',
    'DARK' : 'jwst_miri_dark_0020.rmap',
    'DFLAT' : 'N/A',
    'DISPERSER' : 'N/A',
    'DISTORTION' : 'jwst_miri_distortion_0031.rmap',
    'DRIZPARS' : 'jwst_miri_drizpars_0002.rmap',
    'EXTRACT1D' : 'jwst_miri_extract1d_0011.rmap',
    'FFLAT' : 'N/A',
    'FILTEROFFSET' : 'jwst_miri_filteroffset_0020.rmap',
    'FLAT' : 'jwst_miri_flat_0046.rmap',
    'FORE' : 'N/A',
    'FPA' : 'N/A',
    'FRINGE' : 'jwst_miri_fringe_0014.rmap',
    'GAIN' : 'jwst_miri_gain_0007.rmap',
    'IFUFORE' : 'N/A',
    'IFUPOST' : 'N/A',
    'IFUSLICER' : 'N/A',
    'IPC' : 'jwst_miri_ipc_0005.rmap',
    'LASTFRAME' : 'N/A',
    'LINEARITY' : 'jwst_miri_linearity_0013.rmap',
    'MASK' : 'jwst_miri_mask_0015.rmap',
    'MSA' : 'N/A',
    'MSAOPER' : 'N/A',
    'OTE' : 'N/A',
    'PATHLOSS' : 'N/A',
    'PERSAT' : 'N/A',
    'PHOTOM' : 'jwst_miri_photom_0030.rmap',
    'PSFMASK' : 'jwst_miri_psfmask_0003.rmap',
    'READNOISE' : 'jwst_miri_readnoise_0014.rmap',
    'REFPIX' : 'N/A',
    'REGIONS' : 'jwst_miri_regions_0025.rmap',
    'RESET' : 'jwst_miri_reset_0010.rmap',
    'RESOL' : 'jwst_miri_resol_0004.rmap',
    'RSCD' : 'jwst_miri_rscd_0006.rmap',
    'SATURATION' : 'jwst_miri_saturation_0011.rmap',
    'SFLAT' : 'N/A',
    'SPECWCS' : 'jwst_miri_specwcs_0025.rmap',
    'STRAYMASK' : 'jwst_miri_straymask_0007.rmap',
    'SUPERBIAS' : 'N/A',
    'THROUGHPUT' : 'N/A',
    'TRAPDENSITY' : 'N/A',
    'TRAPPARS' : 'N/A',
    'TSOPHOT' : 'jwst_miri_tsophot_0002.rmap',
    'WAVECORR' : 'N/A',
    'WAVELENGTHRANGE' : 'jwst_miri_wavelengthrange_0019.rmap',
    'WCSREGIONS' : 'jwst_miri_wcsregions_0009.rmap',
    'WFSSBKG' : 'N/A',
}
```

Figure 8-4: MIRI imap

8.3 The RMAP

The rmaps (rules or reference maps) are the heart of the CRDS reference file selection process (https://hst-crds.stsci.edu/static/users_guide/rmap_syntax.html#reference-mappings-rmap, or https://jwst-crds.stsci.edu/static/users_guide/rmap_syntax.html#reference-mappings-rmap). They contain the parameter keywords CRDS uses in reference file selection, as well as the matching rules (combinations of selection keyword values); there could potentially be dozens of matching rules, depending on the number of valid values for a given keyword. The rmaps can also be changed or updated in lieu of resubmitting a reference file (Note that, these updates are handled by the ReDCaT Team via a JIRA issue opened by the instrument team in the CRDS project). For example, if a reference file has an incorrect USEAFTER date, it can be corrected within the rmap using a text editor, and then delivered by the ReDCaT Team; there would be no need to redeliver a corrected file. Note, changes to rmaps ARE NOT reflected in the reference files contained within them. Therefore, when delivering a reference file which replaces another, it's best to check the rmap to ensure nothing is missing or inadvertently added; this is particularly true for cases where one or more of the parkeys has multiple values (e.g., a reference file which has GAIN as a selector, and where that file is applicable to more than one value of the gain - these are separated in the "selector = Match" section of the rmap by or-bars (|)).

Below are examples of HST (Figure 8-5, COS gsagtab) and JWST (Figure 8-6, NIRCam gain) rmaps.

```
header = {
    'derived_from' : 'hst_cos_gsagtab_0255.rmap',
    'filekind' : 'GSAGTAB',
    'instrument' : 'COS',
    'mapping' : 'REFERENCE',
    'name' : 'hst_cos_gsagtab_0256.rmap',
    'observatory' : 'HST',
    'parkey' : (('DETECTOR', 'CENWAVE'), ('DATE-OBS', 'TIME-OBS')),
    'reffile_format' : 'TABLE',
    'reffile_required' : 'NONE',
    'reffile_switch' : 'NONE',
    'rmap_relevance' : '(DETECTOR == "FUV")',
    'sha1sum' : '59e9cc6031b857d4367f8ca1bd4b41a5c96dbca9',
}

selector = Match({
    ('FUV', 'BETWEEN 1055 1097') : UseAfter({
        '2009-05-11 00:00:00' : '23e1646sl_gsag.fits',
    }),
    ('FUV', 'N/A') : UseAfter({
        '2009-05-11 00:00:00' : '41g2040ol_gsag.fits',
    }),
})
```

Figure 8-5: COS gsag rmap

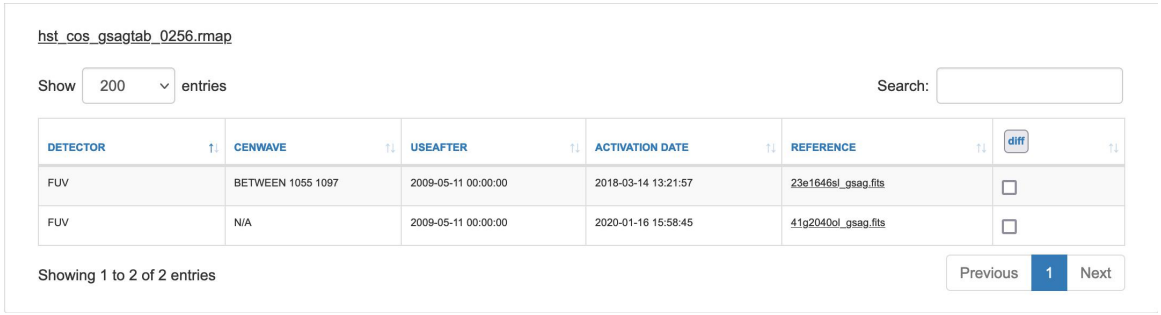
```
header = {
    'classes' : ('Match', 'UseAfter'),
    'derived_from' : 'jwst_nircam_gain_0009.rmap',
    'filekind' : 'GAIN',
    'instrument' : 'NIRCAM',
    'mapping' : 'REFERENCE',
    'name' : 'jwst_nircam_gain_0010.rmap',
    'observatory' : 'JWST',
    'parkey' : (('META.INSTRUMENT.DETECTOR', 'META.SUBARRAY.NAME'), ('META.OBSERVATION.DATE', 'META.OBSERVATION.TIME')),
    'sha1sum' : '67cd41b36ceedcf5672e6a7c19fec1331f397a77',
    'substitutions' : {
        'META.SUBARRAY.NAME' : {
            'GENERIC' : 'N/A',
        },
    },
}

selector = Match({
    ('NRCA1', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0019.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0050.fits',
    }),
    ('NRCA2', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0001.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0052.fits',
    }),
    ('NRCA3', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0020.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0048.fits',
    }),
    ('NRCA4', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0021.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0055.fits',
    }),
    ('NRCALONG', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0022.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0056.fits',
    }),
    ('NRCB1', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0023.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0053.fits',
    }),
    ('NRCB2', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0024.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0049.fits',
    }),
    ('NRCB3', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0025.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0051.fits',
    }),
    ('NRCB4', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0026.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0057.fits',
    }),
    ('NRCBLONG', 'GENERIC') : UseAfter({
        '1900-01-01 00:00:00' : 'jwst_nircam_gain_0027.fits',
        '2015-10-01 00:00:01' : 'jwst_nircam_gain_0054.fits',
    }),
})
```

Figure 8-6: NIRCam gain rmap

As can be seen above, an rmap is separated into two sections: the header, which contains information about the rmap and the reference type it refers to, and the selector, which has the matching rules and the active reference files associated with them.

Below, in figures 8-7 and 8-8, are examples of how the information in the two rmaps above would be seen in the respective instrument and reference sections of the CRDS home page (https://hst-crds.stsci.edu or https://jwst-crds.stsci.edu)

Figure 8-7: Display of COS gsagtab Reference Files

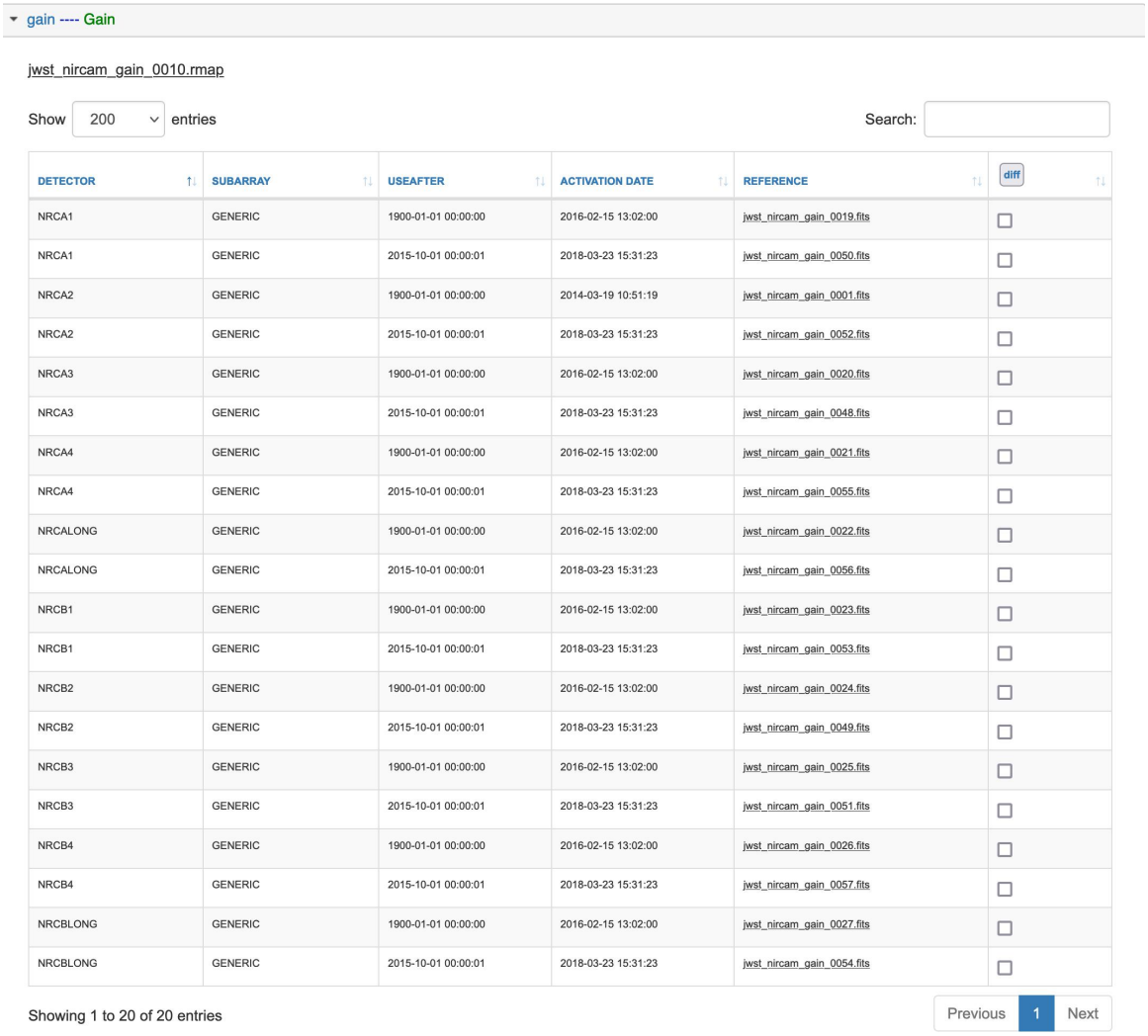


hst_cos_gsagtab_0256.rmap

Show 200 entries Search:

DETECTOR	CENWAVE	USEAFTER	ACTIVATION DATE	REFERENCE	diff
FUV	BETWEEN 1055 1097	2009-05-11 00:00:00	2018-03-14 13:21:57	23e1646sl_gsag.fits	☐
FUV	N/A	2009-05-11 00:00:00	2020-01-16 15:58:45	41g2040ol_gsag.fits	☐

Showing 1 to 2 of 2 entries Previous 1 Next



gain ---- Gain

jwst_nircam_gain_0010.rmap

Show 200 entries Search:

DETECTOR	SUBARRAY	USEAFTER	ACTIVATION DATE	REFERENCE	diff
NRCA1	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0019.fits	☐
NRCA1	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0050.fits	☐
NRCA2	GENERIC	1900-01-01 00:00:00	2014-03-19 10:51:19	jwst_nircam_gain_0001.fits	☐
NRCA2	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0052.fits	☐
NRCA3	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0020.fits	☐
NRCA3	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0048.fits	☐
NRCA4	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0021.fits	☐
NRCA4	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0055.fits	☐
NRCALONG	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0022.fits	☐
NRCALONG	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0056.fits	☐
NRCB1	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0023.fits	☐
NRCB1	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0053.fits	☐
NRCB2	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0024.fits	☐
NRCB2	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0049.fits	☐
NRCB3	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0025.fits	☐
NRCB3	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0051.fits	☐
NRCB4	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0026.fits	☐
NRCB4	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0057.fits	☐
NRCBLONG	GENERIC	1900-01-01 00:00:00	2016-02-15 13:02:00	jwst_nircam_gain_0027.fits	☐
NRCBLONG	GENERIC	2015-10-01 00:00:01	2018-03-23 15:31:23	jwst_nircam_gain_0054.fits	☐

Showing 1 to 20 of 20 entries Previous 1 Next

Figure 8-8: Display of NIRCam gain Reference Files

In the rmap header, filekind is the reference file type, instrument and observatory are self explanatory, and mapping is always set to REFERENCE (for the imap this is set to INSTRUMENT, and PIPELINE for the pmap). The most important entry in the header is 'parkey', which is the set of parameter keywords, which are taken from the data and used by CRDS to select one of the (active) reference files in the 'selector' section. The parkeys are set up differently for each of the telescopes. For HST rmaps, the header keywords are used:

```
'parkey' : (('CCDAMP', 'CCDGAIN', 'CCDOFFST', 'BINAXIS1', 'BINAXIS2'),
('DATE-OBS', 'TIME-OBS'))
```

While for JWST, the data model (metadata) names are used:

```
'parkey' : (('META.INSTRUMENT.DETECTOR', 'META.SUBARRAY.NAME'),
('META.OBSERVATION.DATE', 'META.OBSERVATION.TIME')
```

The parkeys, DATE-OBS and TIME-OBS from an HST dataset, and META.OBSERVATION.DATE and META.OBSERVATION.TIME from a JWST dataset, are combined by CRDS to form a date-time value. This date-time is compared to the UseAfter values in the rmap. The reference file with the UseAfter that is closest to, but greater than, the date-time, will be the one selected, provided the other matching criteria are met.

The 'selector 'area within the rmap contains the matching rules and the active reference files associated with them. The information here can be changed or updated, provided the new values are valid in CRDS for a given reference file type (Note that, during a delivery, CRDS automatically updates the information for a given matching rule). For example, a reference file can be replaced with one that was previously delivered. In the NIRCam gain rmap shown above, the entry

```
('NRCBLONG, 'GENERIC') : UseAfter({
    '1900-01-01 00:00:00' : 'jwst_nircam_gain_0027.fits',
    '2015-10-01 00:00:00' : 'jwst_nircam_gain_0054.fits'
```

could be changed to

```
('NRCBLONG', 'GENERIC') : UseAfter({
    '1900-01-01 00:00:00' : 'jwst_nircam_gain_0027.fits',
    '2015-10-01 00:00:00' : 'jwst_nircam_gain_0044.fits',
```

provided that jwst_nircam_gain_0044.fits had previously been successfully delivered, and there were no other problems with the file.

A new matching rule could be added without having to make a delivery. If, for example, for the COS gsag rmap above, you wanted to add an entry for CENWAVE = 1250 using the same reference file as that for CENWAVE BETWEEN 1055 1097. The rmap would be updated by the ReDCaT Team to become:

```
header = {
    'derived_from' : 'hst_cos_gsagtab_0255.rmap',
    'filekind' : 'GSAGTAB',
    'instrument' : 'COS',
    'mapping' : 'REFERENCE',
    'name' : 'hst_cos_gsagtab_0256.rmap',
    'observatory' : 'HST',
    'parkey' : (('DETECTOR', 'CENWAVE'), ('DATE-OBS', 'TIME-OBS')),
    'reffile_format' : 'TABLE',
    'reffile_required' : 'NONE',
    'reffile_switch' : 'NONE',
    'rmap_relevance' : '(DETECTOR == "FUV")',
    'sha1sum' : '5ecbecf43a9c202f536a4817b39c02319c87f87a',
}
selector = Match({
    ('FUV', 'BETWEEN 1055 1097') : UseAfter({
        '2009-05-11 00:00:00' : '23e1643sl_gsag.fits',
    }),
    (‘FUV’, ‘1250’) : UseAfter({
        ‘2009-05-11 00:00:00’: '23e1643sl_gsag.fits',
    }),
    ('FUV', 'N/A') : UseAfter({
        '2009-05-11 00:00:00' : '41g2040ol_gsag.fits',
    }),
})
```

In the above example, 1250 may or may not be a valid CENWAVE value for the gsag reference file type. As long as a current or successfully delivered previous version of the file is used, CRDS only matches the CENWAVE value in the data to one of the rules in the rmap. It’s only if a new gsag reference file were to be delivered would the CENWAVE value of 1250 need to be known by CRDS. Remember, any changes made to an rmap will not be reflected in the reference file. So while 2331643sl_gsag.fits is associated with CENWAVE=1250 in the rmap, that value will not appear in the header of the file. Therefore, it’s always a good idea to review the rmap before using a reference file as a template.

To have an rmap updated, file a JIRA issue in the CRDS project and assign it to Matt McMaster, and explain in detail what you need done. The ReDCaT Team will then update the file and deliver it. If you wish, you may update the rmap yourself and attach it to the JIRA issue; do not try to deliver the rmap yourself.

Occasionally, you might see a list of parameter values separated by the ‘| ’symbol (or-bar). Examples of this are:

```
('HRC', 'D', '1.0|2.0|4.0|8.0', 'N/A', 1062, 1044, 19.0, 0.0, 'N/A', 'N/A',
'N/A') : UseAfter({
        '1991-01-01 00:00:00' : 'j331211hj_bia.fits',
```

or

```
('NRCA2', 'MASKA210R|SUB640A210R') : UseAfter({
    '2015-06-01 00:00:00' : 'jwst_nircam_dark_0073.fits',
}),
```

For these cases, the given reference file will be used for any of values listed within the set of or-bars. In the ACS example above, that particular bias would be used for data taken after January 1 1991, using the HRC with the D amp, with a gain of 1 or 2 or 4 or 8. The NIRCAM dark would be used for either the MASKA210R or SUB640A210R subarrays. The or-bar values are generated differently depending on the telescope. For HST, they are produced using rules within CRDS when a wildcard value for a keyword is encountered, which is expanded based on instrument, reference file type, and keyword; or-bars may also be added

by updating a given matching rule in the rmap. For example, another way of adding a CENWAVE value of 1250 to the COS gsag rmap for the reference file, 2331643sl_gsag.fits, would be to add an or-bar just after BETWEEN 1055 1097, but still within the quotes. The updated rmap would then look like:

```
header = {
    'derived_from' : 'hst_cos_gsagtab_0255.rmap',
    'filekind' : 'GSAGTAB',
    'instrument' : 'COS',
    'mapping' : 'REFERENCE',
    'name' : 'hst_cos_gsagtab_0256.rmap',
    'observatory' : 'HST',
    'parkey' : (('DETECTOR', 'CENWAVE'), ('DATE-OBS', 'TIME-OBS')),
    'reffile_format' : 'TABLE',
    'reffile_required' : 'NONE',
    'reffile_switch' : 'NONE',
    'rmap_relevance' : '(DETECTOR == "FUV")',
    'sha1sum' : '5ecbecf43a9c202f536a4817b39c02319c87f87a',
}
selector = Match({
    ('FUV', 'BETWEEN 1055 1097|1250') : UseAfter({
        '2009-05-11 00:00:00' : '23e1643sl_gsag.fits',
    }),
    ('FUV', 'N/A') : UseAfter({
        '2009-05-11 00:00:00' : '41g2040ol_gsag.fits',
    }),
})
```

Where 2331643sl_gsag.fits would be selected for COS FUV data where the CENWAVE was between 1055 and 1097 Angstroms, or 1250 Angstroms.

JWST or-values are generated by using P_ keywords within the header of a reference file, where they are given as the value of the keyword. Each value is separated with a bar (‘|’), and a trailing bar is used to designate the end of the entry. Note that each value used must be a valid value known to CRDS.

For the NIRCam dark shown above, jwst_nircam_dark_0073.fits, the following would be in its header to designate that the file is to be used for the MASKA210R or SUB640A210R subarrays:

```
P_SUBARR = MASKA210R|SUB640A210R|
```

When the rmap is generated, the trailing bar is omitted. As long as each of the individual values within the or-bar is valid, CRDS will OK the file. In other words, there is no distinct set values for the P_ keywords that CRDS uses for comparison, it only checks the individual values to ensure they are valid. Note that, during rmap certification, CRDS compares the newly generated rmap to the current version. If any differences are found, a warning will be issued, either “rule change” or “equal weight special case”. If the latter is issued, it needs to be addressed, as it sets up a condition where two (or more) files have equal matching weights, which is a violation of CRDS rules.

You may also see the value of a keyword set to ‘N/A ’within a matching rule. For example, the following is the first matching rule for MIRI dark files, which get selected on DETECTOR, READPATT, SUBARRAY, and USEAFTER:

```
selector = Match({
    ('MIRIFULONG', 'N/A', 'FULL') : UseAfter({
```

```
        '2015-08-01 00:00:00' : 'jwst_miri_dark_0050.fits',
}),
```

In CRDS, ‘N/A ’within a matching rule is known as a ‘weak match’, while a specific value or ANY is known as a ‘strong match ’and would take precedence in reference file selection. In the above example, the ‘N/A ’means that the value of READPATT can be ignored and match only on the values of DETECTOR, SUBARRAY, and USEAFTER.

However, if the filename associated with a matching rule is set to ‘N/A’, CRDS will not search for a reference file and that particular calibration step will be skipped. For example, the NIRCam SPECWCS reference file is selected on PUPIL, MODULE, EXP_TYPE, and USEAFTER. The entry in the rmap for PUPIL=GRISMC, MODULE=A, and EXP_TYPE=NRC_TSGRISM is:

```
('GRISMC', 'A', 'NRC_TSGRISM') : 'N/A',
```

Which means for those data, the reference file is not applicable and the calibration step associated with it will not be performed.

For HST, there are no rmaps where the reference file is given as N/A. Instead if a particular calibration step is not applicable to the data (for example, a step is used only for FUV modes, but the data was taken with the NUV detector), the reference file in the data header will be set to N/A.

Acknowledgements

The author would like to thank Todd Miller, Ed Slavich, and Jonathan Eisenhamer for the information on CRDS they provided, as well as members of the ReDCaT Team for their suggestions and for proofreading this document