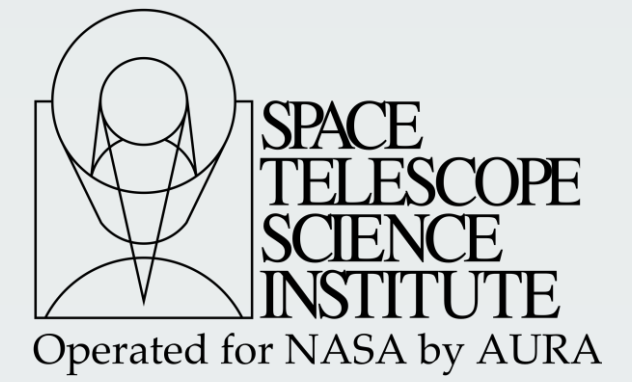




# The JWST Calibration Reference Data System

Howard Bushouse, Perry Greenfield, Todd Miller (STScI)



## Abstract

The James Webb Space Telescope (JWST) Calibration Reference Data System (CRDS) will be used during the instrumental calibration phase of JWST data processing to select reference files (e.g. darks, flats, distortion models) to be used to calibrate each dataset. The CRDS will also be used in the delivery and management of reference files within the JWST data archive and to manage the rules that will be used to determine which files are to be used with each science dataset. The CRDS incorporates lessons learned from more than 20 years of use of the HST Calibration Database System (CDBS), allowing for easier management of the reference file mapping rules, web-based reference file delivery and retrieval procedures, as well as on-line access by off-site users who are reprocessing their science data.

## Purpose of CRDS

### Primary

- Determine which reference files are the most appropriate to be used to calibrate a science dataset

### Secondary

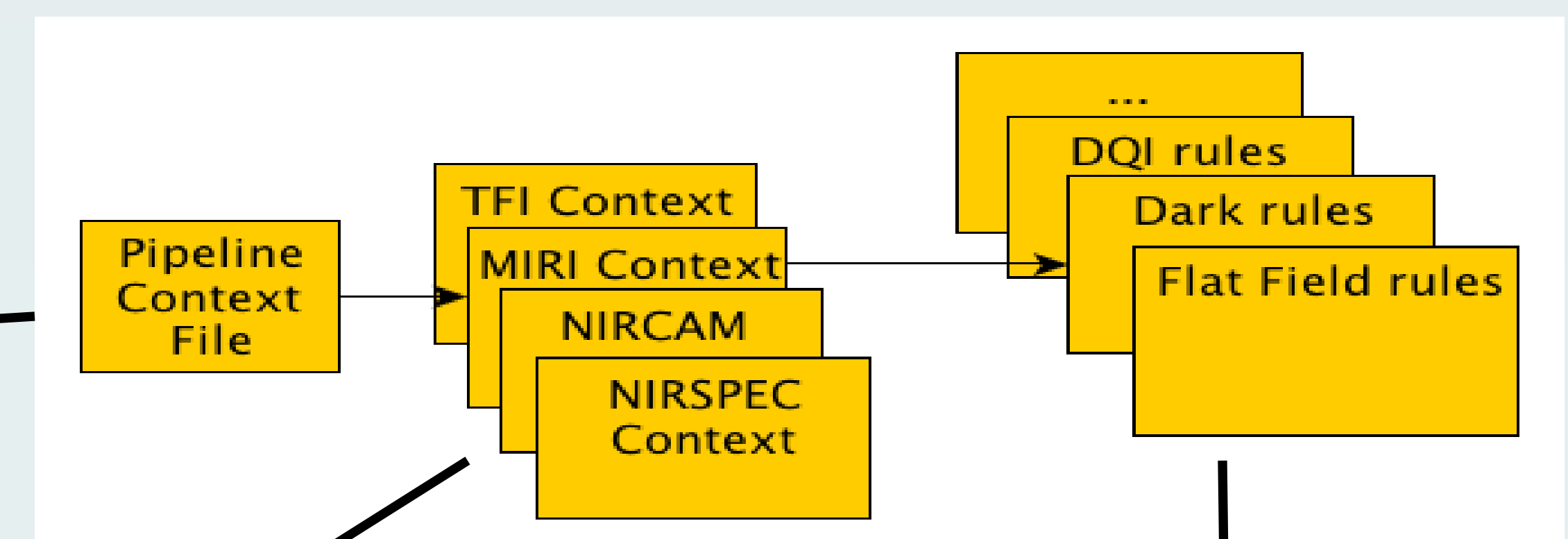
- Manage new reference files and updates to the rules that govern their selection for use with science datasets
- Provide a history of all reference files
- Understand what is currently being used and the effects of changes

## Design Features

- Supports multiple versions of calibration software simultaneously
- Common software interface for both local and remote users
- Remote users can query CRDS through a web interface
- Records meta data about each reference file, including how it was created, who created it, who delivered it, when it was delivered, significance of the changes, etc.
- Provides a tool to show active reference files associated with given observing modes
- Mapping rules are human readable, easy to modify, and extensible
- Calibration pipelines interface directly to CRDS system to get list of reference files to use for a given science dataset (rather than the old HST/CDBS method of populating header keywords)

## Implementation

- Pipeline context file lists instrument context files to use
- Instrument context files list reference file mapping rules files to use
- Mapping rules files list the reference files and selection criteria to use
- No database is used anywhere in the mapping scheme
- All context and mapping rules files implemented as Python dictionaries and lists
- Simple text that is easy to read, understand, and modify



## Pipeline Context File

```

header = {
  'mapping': 'pipeline',
  'observatory': 'HST',
  'parkey': ('INSTRUME',),
}
selector = {
  'ACS': 'hst_acs_010.imap',
  'COS': 'hst_cos_034.imap',
  'STIS': 'hst_stis_004.imap',
  'WFC3': 'hst_wfc3_017.imap',
}
  
```

## Instrument Context File

```

header = {
  'mapping': 'instrument',
  'observatory': 'HST',
  'instrument': 'ACS',
}
selector = {
  'biasfile': ('bia', 'hst_acs_biasfile_217.rmap'),
  'darkfile': ('drk', 'hst_acs_darkfile_115.rmap'),
  'bpixtab': ('bpx', 'hst_acs_bpixtab_003.rmap'),
  'ccdtab': ('ccd', 'hst_acs_ccdtab_005.rmap'),
  'pfltfile': ('pfl', 'hst_acs_pfltfile_013.rmap'),
  ...
  'idctab': ('idc', 'hst_acs_idctab_003.rmap'),
}
  
```

## Mapping Rules File

Matching criteria support numerical or string matches, version number and date inequalities, nearest value, interpolation within a range, and wildcards

### Simple Example for HST/ACS Bias Rules

```

header = {
  'observatory': 'HST',
  'instrument': 'ACS',
  'reftype': 'BIAS',
  'parkey': (('detector', 'gain', 'amp', 'xsize', 'ysize')),
}
selector = Match({
  ('WFC', '1.0', 'A', 2048, 4096): 'cref_hst_acs_bias_0001.fits',
  ('WFC', '2.0', 'A', 2048, 4096): 'cref_hst_acs_bias_0002.fits',
  ('WFC', '4.0', 'A', 2048, 4096): 'cref_hst_acs_bias_0003.fits',
  ('WFC', '1.0', 'A', 1046, 1024): 'cref_hst_acs_bias_0004.fits',
  ('WFC', '1.0', 'A', 1046, 1024): 'cref_hst_acs_bias_0005.fits',
  ...
  ('HRC', '2.0', 'D', 2048, 4096): 'cref_hst_acs_bias_0030.fits',
  ('HRC', '4.0', 'D', 2048, 4096): 'cref_hst_acs_bias_0031.fits',
})
  
```

## Operations Flow

- Pipeline sends request to CRDS (via web service) for names of appropriate reference files for a given science dataset
- CRDS "Bestref" service returns list of names
- Pipeline checks for existence of reference files on local disk
- If they aren't available, request download from CRDS
- Pipeline performs calibration
- Operations do not require reference file names to appear in header keywords, like existing HST/CDBS, so no user updates are needed to use latest/greatest reference files