CRDS to CRDS Pipeline

for JWST

Interface Control Document

2015-08-16

## Overview and Scope

This document describes the shared file system interface used to transfer CRDS files from the CRDS file submission component to the CRDS Pipeline (archive ingest system for reference files). This document does not describe in detail the contents of files or the ways in which CRDS or the CRDS pipeline perform their respective tasks. Instead, it focuses on the exchange of files between CRDS and the CRDS pipeline and their respective obligations and restrictions in complying with the file exchange interface.

## Shared File Directory

Files are delivered from CRDS to the CRDS pipeline by placing them in a shared file delivery directory. First, each reference and map file to be delivered is copied into the shared delivery directory. When the copies are complete, CRDS writes out a file catalog that lists each file included in the delivery, one file per line. After noticing the existence of a new catalog file, the CRDS pipeline ingests the delivered files into the archive. It verifies the files were successfully archived, removes all delivered files and the catalog from the delivery directory, and sends an email indicating the delivery is complete.

### Order of Delivery

CRDS shall deliver reference and rules files first, then write out the catalog that lists the delivered files as a complete set. The creation of the catalog file in the delivery directory signifies that CRDS has completed its part of the delivery process.

### Order of Ingest

The CRDS Pipeline shall ingest catalogs according to the order of sequence numbers, not according to the order of the lexical sort of catalog files. Within each catalog, the order in which the CRDS Pipeline ingests files is undefined. All files are guaranteed to be archived before completion is acknowledged by removing the catalog file.

### Catalog Name and Format

The file manifest (catalog) shall list each delivered file, without a path, one per line. The catalog shall be named matching a regular expression “jwst\_\d+.cat” where the name consists of the string “jwst\_” followed by a sequence number consisting of up to 6 decimal digits followed by the extension “.cat”. The sequence number is not zero-filled so sorting the file names requires parsing the sequence number into an integer value. It is not sufficient to sort based on the unparsed file name strings.

***Example Catalog Name***

An example JWST catalog name is:

jwst\_4.cat

###### Delivery Catalog Example Contents

Catalogs list delivered files one per line:

jwst\_0009.pmap

jwst\_nircam\_0005.imap

jwst\_nircam\_linearity\_0020.rmap

jwst\_nircam\_linearity\_0020.fits

jwst\_nircam\_linearity\_0021.fits

...

##### Acknowledgment of Archiving

The CRDS Pipeline acknowledges the receipt and successful archiving of delivered files by removing each file and the catalog (in all forms) from the shared file delivery directory. Removal of the catalog, including any alternate extensions, signifies to the CRDS server that delivered files are fully in the archive and available and released for distribution and use.

### Users and File Permissions

The CRDS server and delivery subsystem runs as user “crds” and have read-write access to both catalog files and delivered files. The CRDS pipeline accesses the delivery directory under the group “crdsoper” and has the capability of copying, moving, and removing files in the delivery directory only. The CRDS server maintains hard linked references to all files in the delivery area so it is paramount that the CRDS Pipeline not make changes to files (other than renaming) prior to making private copies.

# Delivered File Properties

### Delivered File Naming

Files delivered by CRDS are already named in an official final form that is tracked in the CRDS catalog and associated with file metadata.

Catalog file name and contents

The delivery catalog file lists each delivered file name on a separate line. The CRDS pipeline may rename the .cat file within the delivery directory. The CRDS Pipeline changes the extension of the .cat file to reflect the current stage of pipeline processing:

|  |  |
| --- | --- |
| **Extension** | **Meaning** |
| .cat | Initial list of delivered files from CRDS. |
| .cat\_submit | CRDS pipeline has noticed file delivery and submitted it for processing. |
| .cat\_proc | CRDS pipeline is processing the delivery. |
| .cat\_ERROR | CRDS pipeline has encountered a fatal error which requires troubleshooting. |
| (all forms of catalog removed from delivery area) | All delivered files are in archive ready for distribution. |

Removal of all forms of the .cat file signifies completion of delivery. The CRDS pipeline shall not change the contents of the .cat file. The .cat file name shall be 255 characters or less in length.

R**eferences and mappings**

CRDS mappings and references are delivered with official names in final form. The CRDS pipeline **shall not rename** delivered files. The CRDS pipeline **shall not change the contents** of delivered files in any way.

**Reference and mapping name syntax**

Mapping and reference file names shall be 255 characters or less in length. Mapping and reference file names may consist of any sequence of letters, decimal digits, period, or underscore. The entire file name including extension shall be used to identify the file to the archive. Mapping and reference files are not grouped into sets on the basis of names but are considered individually as unique entities. The mappings and references listed in a single catalog file shall become activated for distribution and use in unison.

### Extra or Extraneous Files

Other than renaming the file delivery catalog, the CRDS Pipeline shall not create temporary files in the file delivery directory. The file extension of the delivery catalog may be changed to signify different states of CRDS Pipeline processing, but the extension shall always contain the sub-string “.cat”.

### File Permissions

A reference, mappings, and catalog files shall be read-only for group “crdsoper”, i.e., the CRDS Pipeline~~)~~. The delivery directory itself shall be rwx to support removal of files by the CRDS pipeline.

### File Size

The archive shall accept reference files up to a length of 16G for a single file. This number is a “best case worst case” estimate and may change depending on calibration software development.

# Pipeline CRDS Cache Synchronization

The current configuration of CRDS for JWST (just like HST) depends on a cache of CRDS files and configuration information that is local to the pipeline. During normal pipeline operations and best references computations~~,~~ the CRDS pipeline runs fully decoupled from the CRDS server and relies on the cache for rules and configuration information. When new files are delivered, several steps are taken to update the pipeline's local CRDS cache.

### First cron\_sync

After successfully archiving the files and acknowledging archival success to CRDS by removing the .cat file, the CRDS Pipeline shall sync delivered files to the pipeline's CRDS cache by running the cron\_sync script. This keeps lengthy file transfer times for reference files as a background process, shieldingpipeline operators from long wait times for their interactive process. The CRDS context does not change at this time; the delivered files become available in the cache but are not yet the pipeline processing default files.

### Set Context

After the CRDS Pipeline performs the first cron\_sync, the CRDS system is ready for the crds\_team and pipeline operator to update the default operational context on the CRDS server using the Set Context page. This changes the central definition of the operational context and makes it available for synchronization to remote pipelines.

### Second cron\_sync

After the pipeline operator has changed the CRDS Server default operational context using Set Context, the operator shall re-execute the cron\_sync script to update the locally cached default context*.* The configuration of cron\_sync differs

### Pipeline Context Switch Verification

This second cron\_sync invocation is nominally called with the switches –verify-context-change and –push-context. --verify-context-change asserts that it is an ERROR if the pipeline's local default context does not change following this cron\_sync. The –push-context switch uploads the default context now defined in the pipeline's CRDS cache to the CRDS server for display in the Remote Caches display, which compares server state to remote cache states.

### No Direct Access to CRDS Cache

The CRDS cache in the pipeline is an integral part of the CRDS design. The CRDS pipeline shall not change the structure, permissions, or contents of the CRDS cache in any way. All routine (i.e., not failure recovery) changes to the CRDS cache in the pipeline shall be performed using the CRDS wrapper script cron\_sync.