

## **JWST FITS Keyword Dictionary**

Christine Boyer

*Space Telescope Science Institute, Baltimore, Maryland, USA;*  
*heller@stsci.edu*

**Abstract.** FITS Header Keywords provide information about the observation, type of data and post-observation processing. Each FITS header contains a unique combination of keywords appropriate for the type of observation. Many keywords are common to all JWST data while others are instrument-specific. The James Web Space Telescope uses JavaScript Object Notation or JSON, to define the collection of keywords appearing in each product's FITS headers. JSON enables a flexible scheme of sharing keyword information across the JWST data reduction and archive systems.

### **1. JWST Keyword Dictionary**

The James Webb Space Telescope (JWST) data reduction pipeline is a Python software suite that automatically processes data taken by the JWST instruments NIRC*am*, NIRSpec, NIRISS, MIRI, and FGS into FITS products. Each FITS header contains a unique combination of keywords that provide information about the observation, type of data and post-observation processing. Many keywords are common to all JWST data while others are instrument-specific. The values for these keywords can originate from the Proposal and Planning System (PPS), Observatory Status File (OSF), Science and Operations Center Project Reference Database (PRD), Science Data Processing (SDP), Calibration Software, engineering data WEB services or telemetry data. The keyword content information for all JWST products are stored in JSON (JavaScript Object Notation), a human readable format for structuring data. JSON facilitates a flexible interface providing keyword content and definitions across data reduction, calibration, archival and documentation systems.

### **2. Implementation in JSON**

The JSON keyword dictionary input files contain all the information needed to populate FITS headers of the data reduction pipeline products. The keyword dictionary relies on a hierarchical use of JSON objects and arrays to allow sharing of keyword definitions and groupings in different FITS products and headers. The ability of JSON to support recursively nested objects and ordered list of values facilitates a logical grouping of keywords according to their use cases.

## 2.1. Mode dependent top level JSON specification

A primary or top level JSON file is assigned to every science instrument and mode. During data receipt the data processing pipeline uses the observation mode to select the proper top level JSON file. The top level JSON files contains objects and arrays of objects which resolve to the specific keyword content needed for an observation. Associated keywords are grouped together in a single object. This ensures a keyword definition need only appear once across all JSON files. These keyword groupings can then be easily moved into FITS header by referring to the object containing their definitions which are relevant to the type of observation (spectroscopic or imaging data, target acquisitions) and instrument.

Excerpt of top level JSON file used for six MIRI modes:

```
"meta" : {
  "title" : "MIRI Imaging Keywords Schema Metadata",
  "type" : "object",
  "properties" : {
    "coordinates" : {
      "title" : "Coordinate information in the file",
      "type" : "object",
      "properties": {"\"$ref\" : \"core.coordinates.schema.json\"}
    }
  },

  "observation" : {
    "title" : "Observation identifiers",
    "type" : "object",
    "instrument" : {
      "title" : "Instrument configuration information",
      "type" : "object",
      "properties" : {
        "allOf" : [
          {"\"$ref\" : \"core.instrument.schema.json\"},
          {"\"$ref\" : \"core.sci_instrument.schema.json\"},
          {"\"$ref\" : \"miri.all.instrument.schema.json\"}
        ]
      }
    }
  }
}
```

## 2.2. Single FITS Keyword JSON definitions

All entries in the top level JSON file eventually resolve into explicit keyword definitions. Individual keyword definitions contain all the relevant documentation and implementation information required to fully specify the keyword. Each processing system is responsible for identifying and implementing those items needed for that processing step. The web service user interface is the only system to use all the keys displaying information for any given keyword. The data production and archival systems use different subsets of the information provided by a single keyword definition. This simple

design allows the users to find all the information about a FITS keyword in one place that is both human readable and easy to parse by computer software.

Table 1. JSON items single FITS keyword specification

JSON key	Required	Use
fits_keyword	Y	FITS 8 character or less keyword name
type	Y	FITS type of keyword value
default_value	Y	Keyword default value
title	Y	FITS comment for the keyword
units	N	Units of value
level	Y	FITS file product
fits_hdu	Y	FITS HDU identification
sw_source	N	Used by data reduction pipeline to populate value
special_processing	N	Designates if keyword is required or optional
comment_line	N	Value precedes keyword as FITS comment line
sql_dtype	N	Archival system field type
destination	N	Archival system table and field
description	N	Long description for documentation
example	N	Example value for documentation

### 2.3. Implementation in Data reduction pipeline JSON

JSON definition of FITS keyword, PROP\_DEC:

```
"proposer_dec" : {
  "title" : "Target proper motion in Dec",
  "type" : "float",
  "sql_dtype" : "float",
  "fits_keyword" : "PROP_DEC",
  "description" : "Target proper motion in declination",
  "special_processing" : "VALUE_REQUIRED",
  "default_value" : "",
  "example" : "0.0",
  "units" : "degrees",
  "sw_source" : "PPS:fixed_target.dec_literal|dms_to_degrees|",
  "source" : "Proposal and Planning System (PPS)",
  "destination":["ScienceCommon.prop_dec","GuideStar.prop_dec"],
  "level" : "1b",
  "si" : "Multiple",
  "section" : "Target Information",
  "fits_hdu" : "PRIMARY",
  "comment_line" : "/ Target information"
}
```

Resulting FITS entry:

```
COMMENT / Target information
PROP_DEC= -66.59798333333333 / Target proper motion in Dec
```

The data reduction pipeline uses required level, fits\_hdu, fits\_keyword, type and title JSON keys to select keywords destined for FITS headers. Keyword content for each FITS header is identified by level, or file type, and FITS header name.

The optional sw\_source item can be used to automatically populate keyword values. The sw\_source identifies the location of the keyword value in either a database or WEB service. A database value is retrieved when the sw\_source field is of the format DATABASE.TABLE.FIELD. The data reduction pipeline builds the where clause for each table according to the unique keys needed for the table. A value from engineering WEB service is retrieved when the sw\_source value is the of URL:mnemonic:start:end. The mnemonic must be a valid engineering item name found in the engineering web service. The start and end values are FITS keywords names whose values provide the time search window. Also the user can specify a python callback to format the value retrieved to the proper FITS type. In the above example, the sw\_source directs the software to retrieve from the PPS database fixed\_target table's field dec\_literal. This database value is then run through the Python callback, dms\_to\_degrees which returns the value formatted properly to a FITS float for the FITS keyword PROC\_DEC.

The special\_processing item defines the keyword as either optional or required for a FITS header. A value of OPTIONAL only adds the keyword to the FITS header if a value has been supplied for the keyword. A setting of VALUE\_REQUIRED stops data reduction pipeline processing with an error if no value is set for the keyword. In the above example processing will stop with error if either no value can be retrieved from the database or the callback's conversion routine fails.

The optional comment\_line item inserts a FITS comment line preceding the FITS keyword in the FITS header. In the above example the comment line "Target information" will be written immediately before the PROP\_DEC entry in the FITS header.

### 3. Summary

The JSON format provides a level of abstraction for the JWST keyword database making this information easily accessible to both users and programming languages. JSON is stored as ASCII text and the key and value notation are self describing. Many programming environments feature the ability to parse and generate JSON. The JSON implementation allows data formats that are not predefined. This flexibility allows the dictionary to grow as new features are added.