

MSSS Original Data Record Mini Header and MSSS Original Data Record Payload Data

Malin Space Science Systems

15253 Avenue of Science, San Diego, CA 92128

mmmdata@msss.com

Release: 2026-01-26

Version: 1.3

Background and Purpose

This document describes the standard for the data transmission and storage formats: MSSS Original Data Record Mini Header and MSSS Original Data Record Payload Data.

The Mastcam, MARDI, and MAHLI scientific imaging instruments onboard the Mars Science Laboratory Curiosity Rover relay data through the rover to ground data systems. This data stream is encapsulated as a file with the assigned extension DAT.

These are called Original Data Record files (ODR files) as the data within are unprocessed and unmodified. ODR DAT files are considered NASA Packet and CODMAC level 1 (Raw) [12]. DAT files begin with a 64 byte MSSS ODR Mini Header followed by MSSS ODR Payload Data. These data shall be described in this document.

EDR DAT and ODR DAT files within PDS3 archives share the same structure and content. This document serves as an extension and clarification of section 4.4.3 in the MSL_MMM_EDR_RDR_DPSIS [13]. Some field titles and definitions have been updated for improved clarity.

Glossary

- MMM - Mastcam, MAHLI, and MARDI science instruments
- DEA - MMM Digital Electronics Assembly (DEA) which controls focus, zoom and filters mechanisms, acquires images, compresses, stores, and interfaces the MMM science instruments to the Mars Science Laboratory (MSL) rover
- ZStack - a series of images with progressive focal plane adjustment
- Companding - signal processing technique that compresses dynamic range before transmission
- CCD - A CCD imager is a semiconductor device that converts photons into electronic charge packets, storing them in individual pixels. These packets are then systematically moved across the chip and converted into a digital signal to form an image
- CFA - Color Filter Array, optical filters as arranged in a repeating pattern on a CCD imaging device
- Bayer pattern - A CFA with a specific arrangement of RGB filters
- Thumbnail - A reduced resolution form of a full sensor resolution image made by subsampling the source image to 1/64 resolution

Constraints and Applicable Documents

This document does not detail image processing or calibration methods. Comprehensive details regarding image processing and calibration are omitted herein, and instead, readers are directed to the appropriate mission documentation. CCD camera system design is beyond the scope of this material. Technical resources are referenced inline for further reading.

[1] JPL. (2023). *Mars Science Laboratory Project Software Interface Specification (SIS), Mast Camera (Mastcam), Mars Hand Lens Imager (MAHLI), and Mars Descent Imager (MARDI) Original Data Record (ODR), Experiment Data Record (EDR) and Reduced Data Record (RDR) PDS Data Products*, JPL D-75410, SIS-SCI035-MSL, version 1.4, April 28, 2023, https://planetarydata.jpl.nasa.gov/img/data/msl/msl_mmm/document/MSL_MMM_DPSIS.PDF, urn:nasa:pds:msl_mmm:document:msl_mmm_dpsis:1.0

[2] Maki, J. N., et al. (2017). *The Mars Science Laboratory (MSL) Mast cameras and Descent imager: Investigation and instrument descriptions*, <https://doi.org/10.1002/2016EA000252>

[3] Bell, J. F., III et al. (2017), *The Mars Science Laboratory Curiosity rover Mastcam instruments: Preflight and in-flight calibration, validation, and data archiving*, *Earth and Space Science*, 4, 396–452, doi:10.1002/2016EA000219

[4] NASA Planetary Data System. (2023). *Mars Science Laboratory Mastcam MAHLI MARDI (MMM) Cameras Bundle*, 2023, <https://doi.org/10.17189/ffgm-vj37>, urn:nasa:pds:msl_mmm::1.0

[5] Edgett, K. S., et al. (2015). *Curiosity's robotic arm-mounted Mars Hand Lens Imager (MAHLI): Characterization and calibration status*, *MSL MAHLI Technical Report 0001 (version 1: 19 June 2015; version 2: 05 October 2015)*. doi:10.13140/RG.2.1.3798.5447 https://planetarydata.jpl.nasa.gov/img/data/msl/msl_mmm/document/MAHLI_TECH_REPT_0001.PDF, <https://doi.org/10.17189/ffgm-vj37>, urn:nasa:pds:msl_mmm:document:mahli_tech_rept_0001::1.0

[6] Edgett, K.S., Yingst, R.A., Ravine, M.A. et al. (2012). *Curiosity's Mars Hand Lens Imager (MAHLI) Investigation*. *Space Sci Rev* 170, 259–317, <https://doi.org/10.1007/s11214-012-9910-4>

[7] ON Semiconductor. (2017). *KAI-2020 1600 (H) x 1200 (V) Interline CCD Image Sensor, Rev 4, Dec 2017*, <https://web.archive.org/web/20250318012139/https://www.onsemi.com/download/data-sheet/pdf/kai-2020-d.pdf>

[8] Maki, J. N., et al. (2003), *Mars Exploration Rover Engineering Cameras*, *J. Geophys. Res.*, 108, 8071, <https://doi.org/10.1029/2003JE002077>, E12, Section 3.1.4

[9] M. Caplinger, Malin Space Science Systems. (2013). *Calibration summary for the MSL Mastcam, MAHLI, and MARDI instruments*, https://planetarydata.jpl.nasa.gov/img/data/msl/msl_mmm/calibration/MSL_MMM_CAL.TXT, urn:nasa:pds:msl_mmm:calibration:msl_mmm_cal

[10] McNair, S. (2023). *MSL MMM Calibration Data* https://planetarydata.jpl.nasa.gov/img/data/msl/msl_mmm/calibration/, <https://doi.org/10.17189/ffgm-vj37>, urn:nasa:pds:msl_mmm:calibration::2.0

- [11] Huffman, D. A. (1952). *A Method for the Construction of Minimum-Redundancy Codes*, [David A. Huffman, Proceedings of the IRE](#) (Volume: 40, [Issue: 9](#), September 1952), <https://doi.org/10.1109/JRPROC.1952.273898>
- [12] NASA Planetary Data System (2022). *PDS4 Processing Levels for Science Data Sets*, February 2, 2022. https://pds.nasa.gov/datastandards/documents/ProcessingLevelsMapping_220202b.pdf
- [13] Mike Malin. (2015). *Mast Camera (Mastcam), Mars Hand Lens Imager (MAHLI), and Mars Descent Imager (MARDI) Experiment Data Record (EDR) and Reduced Data Record (RDR) PDS Data Products*, May 2015, https://planetarydata.jpl.nasa.gov/img/data/mastcam/MSLMST_0014/DOCUMENT/MSL_MMM_EDR_RDR_DPSIS.PDF
- [14] International Telecommunication Union (ITU-T). (1992). *Recommendation T.81: Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines*, <https://www.itu.int/rec/T-REC-T.81-199209-I/en>

Conventions

When referring to data record contents, the size of fields or data stream contents may be specified in bits. The order of bits shall be from highest to lowest significance. For example, a bit stream, aligned on a byte boundary, with fields A of 3 bits and B of 5 bits would be AAABBBBB where AAA are the three most significant bits of the byte.

Integer values within a byte stream are in big endian order where the most significant byte is at lower numbered byte positions. Integers are unsigned unless otherwise noted.

Padding refers to the convention of adding unused storage as spacing so that subsequent data falls on a natural storage boundary such as a byte or word.

MSSS Original Data Record Mini Header

Field Structure

The Mini Header consists of 64 bytes organized as shown in Table 1.

Table 1: Mini Header Field organization

Field	Byte Position	Bits
camera_product_id	0	32
magic_initial	4	32
spacecraft_clock_start	8	32
detector_erase_count	12	16
padding	14	4
CCD_state		4
led1_control	15	1
led2_control		1
led3_control		1
video_readout_mode		1
clkdiv2		1
long_integration_mode		1
test_mode		1
clkdiv1		1
filter_number	16	8
exposure_duration	17	24
first_sample	20	8
first_line	21	8
samples	22	8
lines	23	8
autofocus_init_position	24	15
autofocus_step_size		10
autofocus_step_count		6

Field	Byte Position	Bits
autofocus_zstack		1
auto_exposure_data_cut	28	8
auto_exposure_pixel_fraction	29	8
auto_exposure_percent	30	8
auto_exposure_max_iteration_count	31	8
padding	32	16
image_compression_mode	34	8
image_quality	35	8
padding	36	24
companding_table	39	8
padding	40	1
led_uv		1
led_vis2		1
led_vis1		1
padding		1
hall_filter		1
hall_cover		1
hall_focus		1
instrument_serial_number	41	24
focus_position_count watson focus	44	32
zoom_position_count ACI focus	48	16
filter_position_count	50	16
dc_offset	52	32
initial_size	56	32
magic_final	60	32

Bytes 24-31 of the Mini Header are used for either ZStack fields or image autofocus fields, depending on the product type. The ZStack fields should be used if either of the following are true:

- focus_mode is “ZStack” in the xml label
- GROUP_APPLICABILITY_FLAG=TRUE under ZSTACK_REQUEST_PARMS in the IMG label.

The decision to use either the image autofocus or ZStack field structure can also be made by analyzing the common bytes. A heuristic, based solely on the mini header content, can differentiate between the two field structures. This is possible under the assumption that the values for one structure are mutually exclusive from valid values of the other. A valid autofocus parameter set cannot be misinterpreted as a valid ZStack parameter set, and vice versa. For an implementation of this heuristic, refer to the provided file msss_odr.py.

Table 2: ZStack field organization

Field	Byte Position	Bits
focus_merge_start_camera_product_id	24	32
focus_merge_depth	28	8
padding	29	22
focus_merge_blending_flag		1
focus_merge_registration_flag		1

Field Definitions

camera_product_id

Image products are assigned Camera Data Product IDs (CDPIDs) sequentially, starting at 1. When DEA flash memory is cleared for reallocation of new image products, the first image will again be CDPID 1. The high bit will be set if this is a thumbnail.

magic_initial

An ODR Mini Header shall always contain 0xFF00FOCA in this field.

spacecraft_clock_start

The value of the spacecraft clock is recorded at the start of image acquisition after these events: optical filter movement, focus mechanism movement, and auto exposure.

For focus-merge or range map products, the value is captured at the start of algorithm execution.

The value of SCLK is maintained by a real-time clock in the instrument DEA. This value is synchronized with the spacecraft clock on DEA bootup.

detector_erase_count

CCD charges are cleared using vertical transfers before starting an integration. The number of transfers performed is twice the detector_erase_count value. [7]

led1_control, led2_control, led3_control

Logical values, where 1 indicates that the respective LED remains on after imaging.

video_readout_mode

A logical value of 0 indicates a standard MMM camera imaging mode where the CCD sensor is sequentially cleared, charges accumulate during a specified integration time (exposure), are shifted to vertical CCD registers, and then read line by line as digital values. This is described as Progressive Scan documented on page 23 in the KAI-2020 datasheet [7].

A logical value of 1 indicates that the camera shifts charges out of vertical transfer registers while asynchronously exposing the next frame on the CCD. The readout rate is fixed at 40 MHz.

CCD_state, clkdiv1, clkdiv2

These values designate the CCD readout rate. Their content is ignored in video_readout_mode 1.

clkdiv1	clkdiv2	CCD_state		
		0x0	0xA	0xF
0	0	20 MHz	invalid	invalid
0	1	invalid	invalid	invalid
1	0	5 MHz	3.3 MHz	2.5 MHz
1	1	10 MHz	invalid	invalid

long_integration_mode, test_mode

Used for pre-flight diagnostics.

filter_number

This is the commanded optical filter index for Mastcam instruments. Valid values are 0-7.

exposure_duration

- The CCD integration time is expressed in units of 0.1 ms where 125 indicates 12.5×10^{-3} s.
- Valid exposure values are 0 to 8388
- 0xFFFFFFFF indicates that the previous image exposure value was used
- For autofocused images, this is the exposure time used during acquisition of the focus frame
- For auto-exposures, this is the settled algorithm exposure (actual integration)

first_sample, first_line, samples, lines

- Designate start position and dimensions of image relative to the upper-left CCD origin
- first_sample is sensor column/8 (a value of 2 references sensor column 16)
- first_line is sensor row/8 (a value of 2 references sensor row 16)
- samples and lines are the number of samples/8 and lines/8 from first_sample and first_line (a value of 2 references a sensor sample or a line count of 8 respectively)

Autofocus

These are the commanded autofocus parameters.

- The autofocus algorithm acquires *autofocus_step_count* image frames, with the focus mechanism moving in increments of *autofocus_step_size*. It starts at the *autofocus_init_position*
- If *autofocus_zstack* is 1, then all image frames are retained and collected as image products. These can then be used as inputs to the focus-merge algorithm
- If *autofocus_zstack* is 0, then all frames are discarded except the best-focus image. See footnote reference for autofocus algorithm. [6]
- If *autofocus_init_position* is 0x7FFF, then the initial position becomes the previous focus mechanism position minus *autofocus_step_size* x *autofocus_step_count*/2

Autoexposure

These are the commanded auto-exposure parameters.

- The auto-exposure algorithm[8] acquires image frames starting at the commanded exposure value, adjusting integration time until no more than *auto_exposure_pixel_fraction* (autoexposure fraction in integer percent) of image pixels are higher than *auto_exposure_data_cut*

- Autoexposure will take up to *auto_exposure_max_iteration_count* frames
- Autoexposure will terminate if successive exposure differences are less than or equal to *auto_exposure_percent* or if the exposure time exceeds 10 seconds
- Auto-exposure is not used if all parameters are zero.

image_compression_mode

Compression mode

Value	Compression
0	JPEG grayscale, Y component
1	JPEG with 4:2:2 chroma sampling, YCbCr
2	JPEG with 4:4:4 chroma sampling, YCbCr
0xFF	MSSS lossless
0 with image quality = 0	no compression, RAW raster form

image_quality

If non-zero, this is the JPEG compression quality (valid values 1 to 100).

When both *image_compression_mode* and *image_quality* are 0, there is no compression.

companding_table

There are 33 lookup tables (0 to 32) which compand 12-bit sensor dn values to an 8-bit form.

Lookup tables can be found in PDS4 LID urn:nasa:pds:msl_mmm:calibration [10] .

A value of 0xFF in this field indicates that 12-bit sensor dn values are stored in 16-bit integers and no lookup table is used.

led_uv, led_vis2, led_vis1

A logical value of 1 indicates that the corresponding LED was in an on state during integration.

hall_filter, hall_cover, hall_focus

These are Hall effect sensor flags for the optical filter wheel, MAHLI lens cover, and focus mechanisms. Values are logical with 1 as on.

instrument_serial_number

Each flight DEA has a unique numeric serial number.

focus_position_count

This is a signed integer with the focus motor position in counts for the focus mechanism.

- Empirical motor count to focus distance for the MAHLI imager [5][9]
- Empirical motor count to focus distance for the Mastcam imagers [3]
- This value has no meaning for the fixed focus MARDI imager

zoom_position_count

Reserved

filter_position_count

This is a signed integer containing the optical filter wheel [3] position in counts. Optical filters are aligned on multiples of 294 counts. *filter_position_count/294* indexes the filter on the filter wheel. The value is signed and may be negative. 0 is always the clear filter designated as L0 and R0 as described in [3]. A non-zero residual indicates off-center filter alignment.

dc_offset

The commanded DC offset that was removed from all image dn values before companding.

initial_size

This is the initially allocated product size in flash memory (bytes) which may be reduced during image compression.

magic_final

An ODR Mini Header shall always contain 0x1010CC28 in this field.

ZStack Image Products

These are the commanded parameters to merge the designated in-flash ZStack image products.

If an image sequence is autofocused with *autofocus_zstack* = 1, then *autofocus_step_count* images are acquired and stored in DEA flash memory. An onboard focus-merge algorithm [5,6] can be used later to fuse these images into two new products: a best-focus and a range-map image.

A focus-merge command will specify that the algorithm starts at the flash memory CDPID *focus_merge_start_camera_product_id* and use *focus_merge_depth* images. The corresponding source products, if received and therefore archived, can be inferred.

focus_merge_blending_flag = 1 indicates that blending was performed on the best-focus product.

focus_merge_registration_flag = 1 indicates that ZStack images were registered before merging.

MSSS Original Data Record Payload Data

Payload data may consist of JPEG, losslessly compressed, or uncompressed data.

The payload data always starts at byte offset 64.

JPEG

If *image_compression_mode* is 0, 1, or 2, then the payload contains one or more JPEG compressed images. Each image starts with a JPEG SOI marker 0xFFD8 and ends with JPEG EOI marker 0xFFD9. Group of Picture (GOP) products may contain up to 16 JPEG images.

The JPEG data are compliant with ITU-T Recommendation T.81 and ISO/IEC 10918. Definitions of SOI and EOI can be found in [14].

Predictive Lossless

Companded 8-bit pixel value differences are Huffman [11] encoded. The Huffman table is not in the payload as it is fixed.

Each pixel value will be the previous value added to the next decoded Huffman symbol. Pixels are stored as lines containing width/2 samples. Lines are collected in groups of 4 (a frame). (See Appendix A.) Each frame is prefixed with the sync code 0xFFFF0000.

After all lines are read for a frame, start at the next sync code and continue for the next frame. The next sync code will be aligned on a 32 bit word boundary with 0 padding.

Pixels are demultiplexed to like-kind Bayer color filter array values so there will be four frames: R, G1, G2, and B. The bands can be interleaved into the Bayer pattern and interpolated to RGB pixels if needed. Band frames repeat until the image is populated.

Uncompressed (Raw) Raster

Uncompressed rasters contain lines × samples values. Samples are stored in row major order with no leading or trailing bytes. The first pixel is at byte index 0 and is the upper-left pixel in an image. Pixels are organized as read from the CCD with the Bayer pattern embedded.

For *image_compression_mode* = 0xff, the raster array is 16-bit words per sample, big endian.

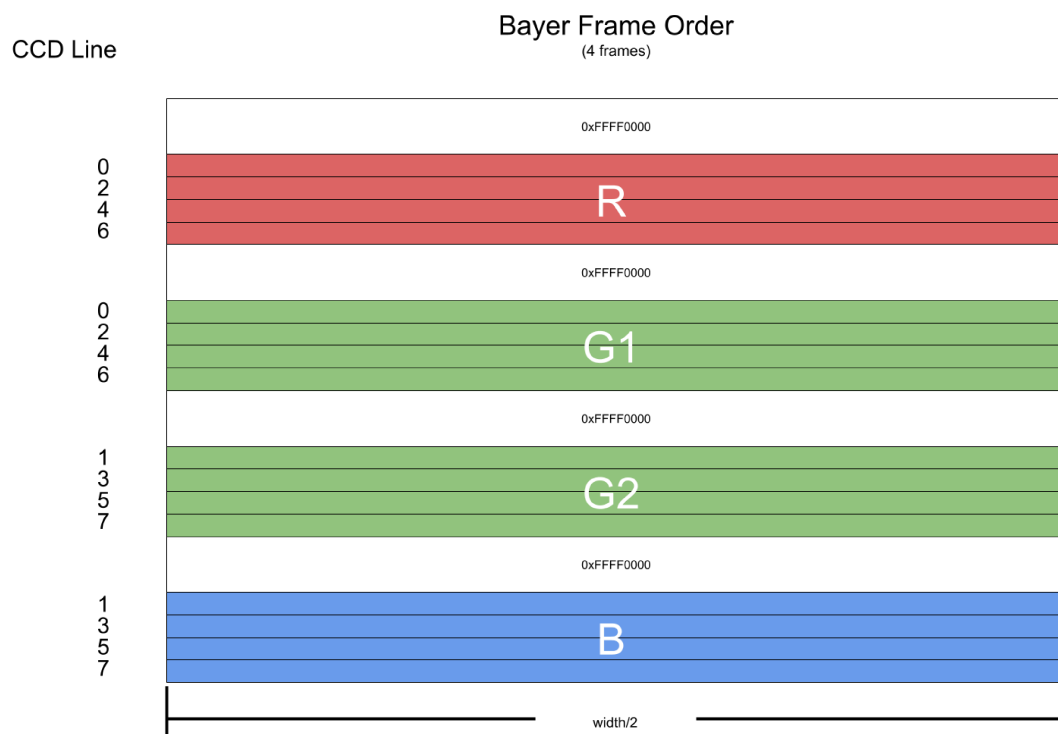
For *image_compression_mode* = 0 and *image_quality* = 0, the array is 8 bits per sample.

Appendix A

Bayer CFA pattern on sensor

Line 0	R	G1				
Line 1	G2	B				
Line 2						
Line 3						

As organized in lossless encoding stream



Appendix B

Reference Software

The software can be found in the MSL MMM Bundle, urn:nasa:pds:msl_mmm:software, under the directory SRC/.

Written in Python, these tools are meant to serve as a reference for other, more performant, implementations.

File	Description
msss_odr.py	Python tool to decode and decompress DAT files
huffman.py	Huffman tables used by msss_odr.py
stream_to_bayer.py	Python tool to interleave line frame output from msss_odr.py into Bayer sensor pixel order

Requirements: Python 3 interpreter, bitstring module, and numpy module

Use pip to install bitstring and numpy

```
pip3 install bitstring numpy
```

msss_odr.py

Usage

```
msss_odr.py filename.dat
```

Expected output(s)

- For JPEG payloads, one or more JPEG files will be written named img_<n>.jpg.
- For predictive lossless and uncompressed raster payloads, a headerless file containing all pixel values, ordered as defined for uncompressed raster payload above: named either img_8bit.raw or img_16bit.raw

stream_to_bayer.py

Usage

```
stream_to_bayer.py infile outfile width height
```

Expected output

Output for losslessly encoded images can be reformed to Bayer sensor order. This is useful for interpolation.

- A headerless file containing pixels in CCD natural order.

Examples

Decode DAT with single JPEG

https://planetarydata.jpl.nasa.gov/img/data/msl/MSLMRD_0001/DATA/EDR/SURFACE/0000/0000MD0000000000100034E01_XXXX.DAT

```
msss_odr.py 0000MD0000000000100034E01_XXXX.DAT

camera_product_id          34
magic_initial              ff00f0ca
spacecraft_clock_start     397501995
detector_erase_count       2047
CCD_state                  0
led1_control               0
led2_control               0
led3_control               0
video_readout_mode         0
clkdiv2                    0
long_integration_mode      0
test_mode                  0
clkdiv1                    0
filter_number              0
exposure_duration          9
first_sample               0
first_line                 0
samples                    206
lines                      150
focus_merge_start_camera_product_id 0
focus_merge_depth         0
focus_merge_blending_flag 0
focus_merge_registration_flag 0
image_compression_mode     1
image_quality              85
```

```

companding_table          0
led_uv                    0
led_vis2                  0
led_vis1                  0
hall_filter               0
hall_cover                0
hall_focus                0
instrument_serial_number  3001
focus_position_count     0
zoom_position_count       0
filter_position_count     0
dc_offset                 155
intial_size               1977664
magic_final               1010cc28
Dimensions 1648x1200
Writing JPEG frame 0 to img_0.jpg

```

Decode file with lossless encoding stream

https://planetarydata.jpl.nasa.gov/img/data/msl/MSLMRD_0002/DATA/EDR/SU RFACE/0000/0000MD0000000000100034C00_XXXX.DAT

```

msss_odr.py 0000MD0000000000100034C00_XXXX.DAT

camera_product_id        34
magic_initial             ff00f0ca
spacecraft_clock_start   397501995
detector_erase_count     2047
CCD_state                0
led1_control              0
led2_control              0
led3_control              0
video_readout_mode       0
clkdiv2                  0
long_integration_mode     0
test_mode                0
clkdiv1                  0
filter_number            0
exposure_duration        9
first_sample             0
first_line               0
samples                  206
lines                    150
focus_merge_start_camera_product_id 0
focus_merge_depth       0
focus_merge_blending_flag 0
focus_merge_registration_flag 0
image_compression_mode    255

```

```
image_quality          0
companding_table       0
led_uv                 0
led_vis2               0
led_vis1               0
hall_filter            0
hall_cover             0
hall_focus             0
instrument_serial_number 3001
focus_position_count  0
zoom_position_count    0
filter_position_count  0
dc_offset              155
intial_size            1977664
magic_final            1010cc28
Dimensions 1648x1200
Writing lossless to img_8bit.raw
```

Interleave to Bayer pattern

```
stream_to_bayer.py img_8bit.raw bayer.raw 1648 1200
Image processed and saved to "bayer.raw"
```

The users are directed to a library such as Pillow or imageio to convert “bayer.raw” to a common image format.

For command line conversion, ImageMagick can be used as follows

```
convert -depth 8 -size 1648x1200 gray:bayer.raw bayer.png
```

Appendix C

Huffman Decompression Algorithm

As demonstrated in the source code provided, the algorithm for node traversal and value retrieval is described below. The Huffman tables are included in the file `huffman.py`.

get_symbol()

Byte symbols are collected from the payload bit stream, a byte at a time, using the traversal table to navigate tree branches. Symbols are added to the previous value for difference coding. The value is an 8-bit register that wraps on overflow

Symbols are formed by traversing the Huffman tree. There is one table per branch, left and right. The traversal table directs node advancement.

Starting at node 0, the stream is decoded a bit at a time. A logical bit value of 0 indicates that the left table node should be used. A value of 1 indicates the right. When left, a traversal node with bit 1 set (10_2) indicates a branch; 0 indicates a leaf. When right, bit 0 (01_2) in the traverse node is used instead to indicate a branch. When reaching a leaf, the byte symbol returned is then the present value of the node in the current table.