Phoenix (PHX)

Software Interface Specification

Interface Title: **Camera Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products**

Mission: PHX   Date: Apr. 20, 2009

Module ID: PHX-274-327

Module Type (REFerence Only or MISsion-specific info included): MIS

Reference Module ID: N/A   Date: N/A

Signatures

GDS Generating Elements:

Ops Product Generation Subsystem (OPGS)
Payam Zamani

Subsystem Engineer   Date

GDS Receiving Elements:

PHX Science Manager
Leslie Tamppari

Manager   Date

Concurrance:

SSI Instrument Lead
Mark Lemmon

Instrument Lead   Date

RAC Instrument Lead
Uwe Keller

Instrument Lead   Date

MECA-OM Instrument Lead
Michael Hecht

Instrument Lead   Date

RSVP Development Team
Brian Cooper

Cognizant Engineer   Date

PSI Development Team
Jason Fox

Cognizant Engineer   Date

GDS Manager
Marla Thornton

Subsystem Engineer   Date

PDS Program Manager
Edwin Grayzeck

Manager   Date

PDS Discipline Node Manager
Sue Lavoie

Manager   Date

Phoenix Project

Software Interface Specification (SIS)

Camera
Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products

Version 1.1.2

Custodians:

Doug Alexander, Robert Deen, Payam Zamani

Paper copies of this document may not be current and should not be relied on for official purposes. The current version is in the Community Server at https://community.jpl.nasa.gov/_layouts/tcclogin/login.aspx?ReturnURL=%2fphx%2fprojects%2fPhoenix%2fexternalus%2fJPL%2520Team%2520Work%2520Area%2520Collection%2fForms%2fAllItems.aspx%3fRootFolder%3d%252fphx%252fprojects%252fPhoenix%252fexternalus%252fJPL%2520Team%2520Work%2520Area%2520Collection%252f07%2520Mission%2520System%252fMOS%252fGDS%252fOPGS%252dMIPL%26View%3d%257b8C02C2C0%252dE927%252d48A4%252dA0A7%252dE1B8FA7065DE%257d

JPL D-33231

April, 2009

Jet Propulsion Laboratory
California Institute of Technology

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
1/9/07	<u>Section 4</u> • Added SSI Camera CCD Readout diagram as Figure 4.	New	Version 1.03
2/10/07	<u>Appendix A</u> • Removed label Groups RA_COORDINATE_SYSTEM, RAC_COORDINATE_SYSTEM, SSI_COORDINATE_SYSTEM • Removed "_ID" string from keywords OPS_TOKEN_ID, OPS_TOKEN_PAYLOAD_ID, OPS_TOKEN_COMMAND_ID, OPS_TOKEN_ACTIVITY_ID • Removed keywords MODEL_COMPONENT_7, MODEL_COMPONENT_8, MODEL_COMPONENT_9 • Removed keyword INST_CMPRS_SEGMENT_QUALITY <u>Appendix B</u> • Removed "_ID" string from keywords OPS_TOKEN_ID, OPS_TOKEN_PAYLOAD_ID, OPS_TOKEN_COMMAND_ID, OPS_TOKEN_ACTIVITY_ID • Removed keywords MODEL_COMPONENT_7, MODEL_COMPONENT_8, MODEL_COMPONENT_9 • Removed keyword INST_CMPRS_SEGMENT_QUALITY	Corrections	Version 1.04
3/29/07	<u>All Sections</u> • Corrected references ("[Ref n]") mapping throughout body of document. • Changed "ROVER_COORDINATE_SYSTEM" to PAYLOAD_COORDINATE_SYSTEM" globally. <u>Section 3.2.1.1</u> • Corrected Figure 3.2.1.1 <u>Section 3.2.1.2</u> • Corrected text describing handling of 8-bit data for EDR generation. <u>Section 4.2.1</u> • Swapped the "Eye" and "Filter" fields in the Single-frame EDR/RDR filename convention. <u>Section 4.2.2</u> • Inserted entire Mosaic filename nomenclature and field descriptions <u>Section 4.2.3</u> • Inserted entire Terrain Mesh filename nomenclature and field descriptions <u>Section 5.2.3.1</u> • Added this section under Section 5.2.3	Corrections	Version 1.06

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
	(Disparity RDR) to describe "Stereo Pair Matching" method used by MIPL. Section 6.1 • Corrected text referring to "Section 3.2.1" to refer to "Section 4.2". Table 5.2 • Corrected number of bands value for Reachability product Appendix B • Corrected Type and Units for ARTICULATION_DEVICE_COUNT • Added valid values for INST_CMPRS_SEGMENTS • Added valid values for MECA-OM instrument for DATA_SET_ID and DATA_SET_NAME • Changed 2nd instance of phrase "Azimuth position at time of …" to "Elevation position at time of …" in Description for keyword ARTICULATION_DEVICE_ANGLE		
4/23/07	Section 4.2.1 • Modified description for "pay" field to show all 16 permutations of Bit String	Corrections	Version 1.06
7/6/07	All • Added diagrams, including Coordinate System, and more text (MIPLRAD Rad-correction & Coord System) in prep for 1st Signature cycle	Corrections	Version 1.0
9/4/07	Signature Page • Added "Edwin Grayzeck" Appendix B • Corrected SSI Eng. Model valid values for ARTICULATION_DEV_POSITION, ARTICULATION_DEV_POSITION_ID, and FILTER_NAME, changing "SSI_L4_700NM" to "SSI_L4_OPEN", "SSI_L8_700NM" to "SSI_L8_D_OPEN", "SSI_R4_700NM" to "SSI_R4_OPEN", and "SSI_R8_700NM" to "SSI_R8_D_OPEN". • Modified definitions for INSTRUMENT_AZIMUTH and INSTRUMENT_ELEVATION	Corrections	Version 1.0
10/26/07	All • Changed instances of string "rover" to "lander" where applicable. Sections 2.1, 2.2 & 2.3 • Inserted edits received from Roger Tanner for Tables 2.1.1, 2.1.2, 2.2.1 and 2.3 Appendix B • Corrected definitions and Valid Values and Source values for INSTRUMENT_TEMPERATURE_COUNT and INSTRUMENT_TEMPERATURE_NAME • Inserted new keyword LED_BITMASK	Corrections	Version 1.0

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
1/8/08	<u>Sections 2.3, 5.2, 6.3</u> • Inserted edits received from Mike Hecht TBD's pertaining to MECA-OM in these Sections <u>Appendix B</u> • Updated Valid Values for MECA-OM (per Mike Hecht) for INSTRUMENT_TEMPERATURE_COUNT and INSTRUMENT_TEMPERATURE_NAME	Corrections	Version 1.0
3/10/08	<u>Sections 2.3, 5.2, 6.3</u> • Inserted edits received from Mike Hecht TBD's pertaining to MECA-OM in these Sections. <u>Appendix B</u> • Updated Valid Values for MECA-OM (per Mike Hecht) for INSTRUMENT_TEMPERATURE_COUNT and INSTRUMENT_TEMPERATURE_NAME.	Corrections	Version 1.0
3/19/08	<u>Section 1</u> • Cleaned up terms for SSI, RAC and MECA-OM team references in Table 1.3.1. <u>Section 4.4</u> • Added "XMS" and "XML" 3-char descriptions for "Masked XYZ RDR" product types in Product Type tables for RDR filenames (single and Mosaics). <u>Section 5.2.1</u> • Inserted Bob Deen's rewrite of Rad-correction text, including new formulas for the MIPL methods (MIPLRAD, MIPLRAD2, MIPLRAD3). <u>Section 5.2.1.2</u> • Cleaned up text, resolved TBD's. <u>Section 5.2.8</u> • Added note that Terrain Meshes do not contain PDS-compliant labels and will not be archived. <u>Section 6.3</u> • Removed MECA-OM coordinate frame from Table 6.3. <u>Appendix A</u> • Reviewed and fixed keyword valid value examples. • Fixed keyword-to-Instrument mapping for EDR keywords. <u>Appendix B</u> • Resolved all TBD's with keywords (definitions, valid values, sources) EXCEPT for the following: PIXEL_DOWNSAMPLE_OPTION, RADIANCE_OFFSET, RADIANCE_SCALING_FACTOR (need Mark Lemmon's input for closure of these). <u>Appendix C</u> • Deleted this Appendix.	Corrections	Version 1.0
3/28/08	<u>Section 5.2.1</u> • Inserted Mark Lemmon's comments received	Corrections	Version 1.0

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
	3/27 to resolve TBD's in Rad-correction test. Appendix B • Inserted Mark Lemmon's comments received 3/27 to resolve issues with descriptions for PIXEL_DOWNSAMPLE_OPTION, RADIANCE_OFFSET and RADIANCE_SCALING_FACTOR.		
4/10/08	Section 4.4 • Inserted "RSD/RSL" and "RUF/RUL" into list of 3-char Product Type identifiers for filenames. Section 5.2.7 • Added this section for Surface Roughness RDR. Appendix B • Re-inserted keyword SEQUENCE_ID to accommodate PSI software needs.	Corrections	Version 1.0
4/14/08	Cover Page • Replaced "Stubbe Hviid" with "Uwe Keller" as RAC Instrument Lead. Section 5.2.1.2 • Inserted text from Stubbe Hviid for description of RAC radiometric correction.	Corrections	Version 1.0
12/08	Resolved all TBDs and updated all sections per discussions from PAWG and email feedback	Correction	V 1.1.1
4/20/09	Appendix A Inserted DATASET_ID and DATASET_NAME for the SSI IOF "Science" RDR dataset. Corrections were made to all DATASET_SET_NAME valid values by including the CODMAC level number component and removing the "-RDR-" component. [R. Alanis]	Correction	V 1.1.2

OPEN ACTION ITEMS

ITEM	ASSIGNEE

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
CCD	Charged Coupled Device
EDR	Experiment Data Record
FEI	File Exchange Interface
FSW	Flight Software
GDS	Ground Data System
GSE	Ground Support Equipment
ID	Identification
JPL	Jet Propulsion Laboratory
LS	Limit Switch (MECA-OM)
MECA	Microscopy, Electrochemistry, and Conductivity Analyzer
MIPL Multimission Instrument Processing Laboratory	Multimission Instrument Processing Laboratory
MIPL	
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
OM	Optical Microscope
OPGS	Operations Product Generation Subsystem
PDS	Planetary Data System
PSI	Phoenix Science Interface
RA	Robotic Arm
RAC	Robotic Arm Camera
RDR	Reduced Data Record
RSVP	Rover Sequencing and Visualization Program
SCLK	Spacecraft Clock
SFDU	Standard Format Data Unit
SIS	Software Interface Specification
SOC	Science Operations Center
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events kernels
SSI	Surface Stereoscopic Imager
SWTS	Sample Wheel and Translation Stage
TBD	To Be Determined/Defined
TDS	Telemetry Delivery Subsystem

VICAR Video Image Communication and Retrieval

GLOSSARY

TERM	DEFINITION

1. INTRODUCTION

1.1 Purpose and Scope

The purpose of this Data Product Software Interface Specification (SIS) is to provide consumers of PHX Camera Payload Experiment Data Record (EDR) and Reduced Data Record (RDR) operations data products with a detailed description of the products and how they are generated, including data sources and destinations. Customers of this SIS include developers and users of the Operations Product Generation Subsystem (OPGS), the Activity Planning and Sequencing Subsystem (APSS) that includes the Rover Sequence and Visualization Program (RSVP) and Phoenix Science Interface (PSI) tools, science instrument team tools, and project-associated scientists and others in the general planetary science community who will analyze the data.

There are three PHX imaging instruments discussed in this document: 1) Surface Stereoscopic Imager (SSI), 2), Robotic Arm Camera (RAC) and 3) MECA Optical Microscope (MECA-OM). Though all three instruments share the similar electronics design and spacecraft interfaces, they differ in the optics, mounted positions, and articulation methods. MECA-OM is characterized by optical and radiometric properties that substantially differ from those of SSI and RAC.

The EDR data product is the raw, uncalibrated, uncorrected image data acquired by the PHX camera instrument. For the three PHX camera instruments, the EDR is identical in format, except for some product label differences. As such, subsequent references to the term EDR in the remainder of this document imply the general raw data product for all three of the aforementioned PHX camera instruments, unless otherwise specified.

Generation of the EDR data product includes methods of decompression to restore data that were lossy or losslessly compressed onboard. Data that were subjected to onboard “12 to 8-bit” scaling are inversely re-scaled as part of the EDR ground data processing.

The RDR data product is derived directly from one or more EDR or RDR data products, and is comprised of radiometrically decalibrated and/or camera model corrected and/or geometrically altered versions of the raw camera data.

1.2 Contents

This Data Product SIS describes how the EDR data product is acquired by the camera and how it is processed, formatted, labeled, and uniquely identified, and how the RDR data product is derived from EDR or RDR data products. The document discusses standards used in generating the product and software that may be used to access the product. The EDR and RDR data product structure and organization is described in sufficient detail to enable a user to read the product. Finally, examples of composite EDR/RDR labels are provided, along with the definitions of the keywords in the label.

1.3 Constraints and Applicable Documents

This SIS is meant to be consistent with the contract negotiated between the PHX Project and the three PHX camera instrument Principal Investigators (PI) in which reduced data records and documentation are explicitly defined as deliverable products. Because this SIS governs the specification of camera-related products used during operations, any proposed changes to this SIS must be impacted by all affected software subsystems observing this SIS in support of operations (e.g., APSS, OPGS).

Secondly, keywords may be added to future revisions of this SIS. Therefore, it is recommended that software designed to process EDRs and RDRs specified by this SIS should be robust to (new) unrecognized keywords.

Additionally, this Data Product SIS is responsive to the following PHX documents:

1. Phoenix Project Archive Generation, Validation and Transfer Plan, JPL D-29392, Rev. 1.0, December 20, 2004.
2. Mars Exploration Program Data Management Plan, R. E. Arvidson and S. Slavney, Rev. 3.0, March 20, 2002.

Additionally, this SIS is also consistent with the following Planetary Data System documents:

3. Planetary Data System Data Standards Reference, JPL D-7669, Version 3.7, Part 2, March 20, 2006.
4. Planetary Data System Archive Preparation Guide, Version 0.050503, May 3, 2005.
5. Planetary Data System Data Standards Reference, JPL D-7669, Version 3.3, Part 2, June 1, 1999.
6. Planetary Data System Data Preparation Workbook, JPL D-7669, Version 3.1, Part 1, February 1, 1995.

Finally, this SIS makes reference to the following documents for technical background information:

7. Pointing, Positioning, Phasing & Coordinate Systems Master (PPPCS), S.R. Doudrick, JPL D-19720, June 28, 2001.
8. A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras, Y. Yakimovsky and R. Cunningham, January 7, 1977.
9. Camera Calibration, D. Gennery, JPL IOM 347/86/10, February 5, 1986.
10. Sensing and Perception Research for Space Telerobotics at JPL, D. Gennery et al., *Proceedings of the IEEE Intern. Conf. on Robotics and Automation,* March 31 - April 3, 1987.
11. Camera Calibration Including Lens Distortion, D. Gennery, JPL D-8580, May 31, 1991.
12. Algorithm for Using CAHV to Determine SGI Graphics Viewpoint and Perspective, B. Bon, JPL IOM 3472-91-057, August 6, 1991.
13. Inclusion of Old Internal Camera Model in New Calibration, D. Gennery, JPL IOM 386.3-94-001, February 22, 1994.
14. “Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points”, Calibration and Orientation of Cameras in Computer Vision, D. Gennery, ISBN 3-540-65283-3, 2001.
15. Computations for Generalized Camera Model Including Entrance, Part 1 and Part 2, D. Gennery, unpublished, May 23, 2001.
16. Generalized Camera Calibration Including Fish-Eye Lenses, D. Gennery, JPL D- 03-0869, 2002.
17. Issues with Linearization, R. Deen, JPL Docushare Collection 2700, File 75670, 2003.
18. Planetary Science Data Dictionary Document, Rev E, JPL D-7116, August 28, 2002.
19. MER IMG Flat Field Scaling, J. Maki, JPL Docushare, File 26302, March 15, 2002.

1.3.1 Relationships with Other Interfaces

Changes to this EDR/RDR data product SIS document affect the following products, software, and/or documents.

Table 1.3.1 - Product and Software Interfaces to this SIS

Name	Type P = product S = software D = document	Owner
MIPL database schema	P	MIPL (JPL)
PHX Camera Payload EDRs • SSI • RAC • MECA-OM	P	MIPL (JPL)
SSI RDRs	P	MIPL (JPL), SSI Science Team
RAC RDRs	P	MIPL (JPL), RAC Science Team
MECA-OM RDRs	P	MECA-OM Science Team
PHXTELEMPROC	S	MIPL (JPL)
RSVP	S	RSVP Dev Team (JPL)
PSI	S	PSI Dev Team (JPL)
SSI Software • TAMCAL • SSITOOLS (SSISPEC, PHXMAP2, MERSTAMPS, MERVIEW)	S	Texas A&M University
RAC Software • RACCAL	S	MPS
MECA-OM Software • RACCAL • Fairwood PDS Image Viewer	S	• Texas A&M University • MECA-OM Science Team (S. Hviid)
Mars Program Suite • MARSCAHV • MARSRAD • MARSJPLSTEREO • MARSCOR3 • MARSXYZ • MARSUVW • MARSRANGE • MARSREACH • MARSROUGH • MARSMAP • MARSMOS • MARSMCAULEY • MARSNAV	S	MIPL (JPL)

Name	**Type** **P = product** **S = software** **D = document**	**Owner**
• MARSTIE • MICA • SUMMITT/CRUMBS		

2. INSTRUMENT OVERVIEW

The PHX camera instrument payload is comprised of three camera systems operating four individual cameras that each differ in the optics, mounted position, and articulation methods. The SSI is the only stereo camera system.

Of the four cameras, there is one set of stereo pairs and two single cameras, as listed in Table 2. Camera mounting locations are shown in Figure 2.

Table 2 - Tabulation of PHX Cameras

PHX Camera Payload Instrument	Location	Number of Cameras
Surface Stereoscopic Imager (SSI)	Stereo pair on Mast Assembly	2
Robotic Arm Camera (RAC)	Robotic Arm (RA)	1
Microscopy, Electrochemistry, and Conductivity Analyzer Optical Microscope (MECA-OM)	inside MECA instrument	1
	TOTAL	4

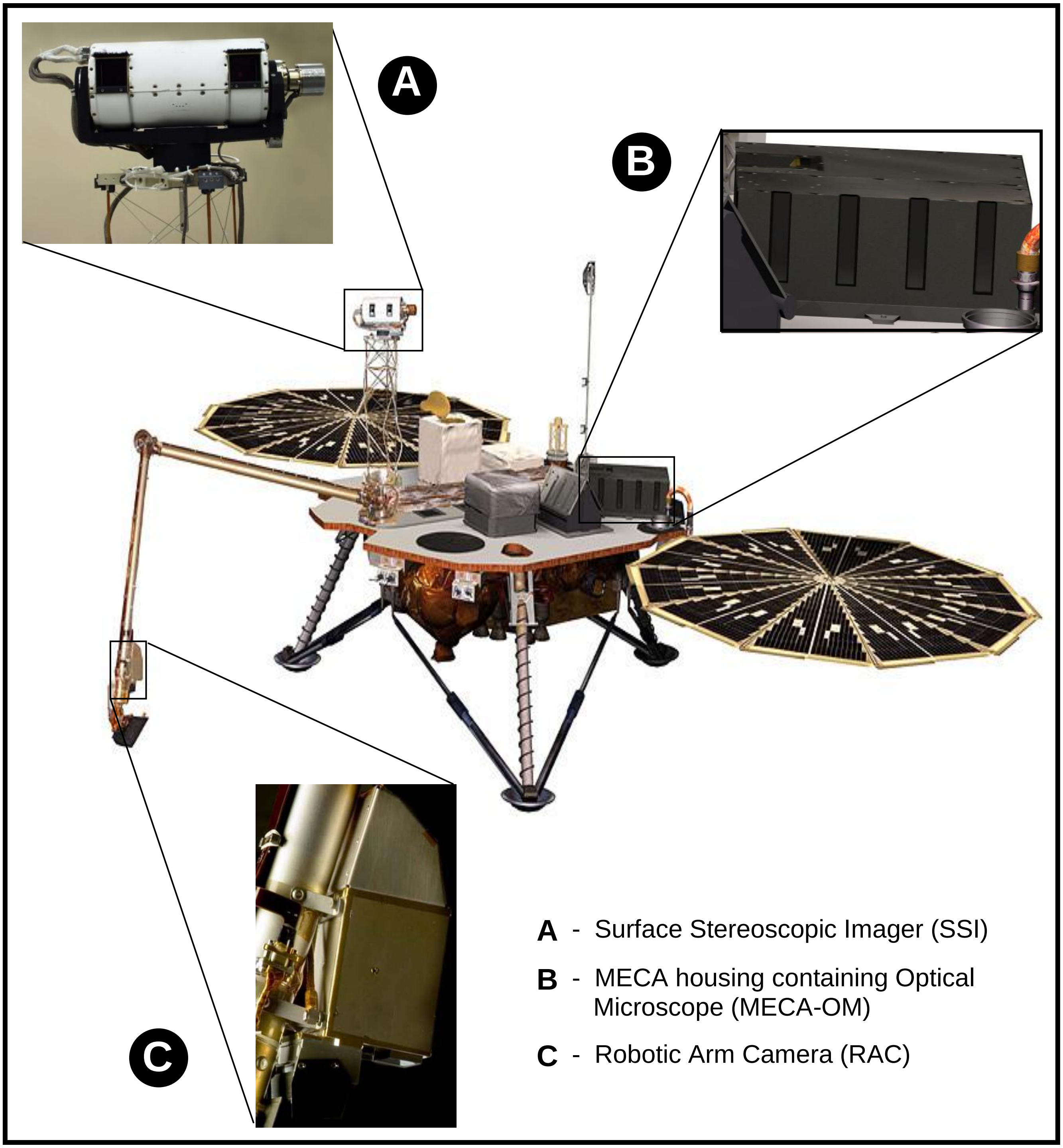


Figure 2 - PHX Camera Payload Configuration

Because of the electronics commonality, image data from all cameras are functionally equivalent and are treated identically in Flight Software (FSW) and the Ground Data System (GDS).

The three PHX camera instruments are discussed in general terms in the following subsections.

2.1 Surface Stereoscopic Camera (SSI)

The SSI is a stereo pair of science cameras at 15.09 cm baseline separation (23.96 mrad average toe-in for right and left) mounted at the top of the mast with a range of motion between 89.8 deg and –92.1 deg elevation and 355.4 degrees of azimuth (with 4.6 deg dead zone). Narrow-angle optics provide an angular resolution of 0.24 mrad/pixel. The SSI will be used to image the surface and sky of Mars around the landing site. The images will be used primarily for Science analysis. SSI images will also support targeting of trench digs by instruments on the RA, providing larger scale perspectives of the work areas than the RA-mounted RAC imaging instrument.

SSI optics characteristics useful in the analysis of EDR and RDR products are described in Table 2.1.1 below:

Table 2.1.1 - SSI Optics Characteristics

Characteristic	Value
Field of View (FOV)	Left: 13.88 deg x 13.88 deg Right: 13.74 deg x 13.74 deg
Baseline Stereo Separation	15.09 cm, 50 mrad toe-in (25 ± 0.5 mrad per eye)
Angular Resolution	Left: 0.2375 mrad/pixel Right: 0.2353 mrad/pixel
Spectral Bandpass	445 nm – 1001 nm
Number of Spectral Filters	12 per camera

SSI has a dual 12-position filter wheel. For the Flight Model, 15 of the 24 filters provide color imaging capability in 13 unique wavelengths from 445 to 1001 nm, two provide color at deck focus, one provides polarization at 750 nm, and the remaining six have neutral density coatings to provide direct solar imaging capability. The spectral bandwidths are described for the Flight Model in Table 2.1.2 and for the Engineering Model in Table 2.1.3 below:

Table 2.1.2 - SSI Spectral Filters (Flight Model)

Filter Number	Left Eye Center Wavelength, nm	Right Eye Center Wavelength, nm	Application
1	672 ± 2	672 ± 2	Surface Stereo pair (RED)
2	445 ± 2	445 ± 2	Surface Stereo pair (BLUE)
3	451 ± 6	671 ± 6	Neutral Density Atmosphere
4	990 ± 0.6	935.9 ± 0.6	Neutral Density Water Vapor
5	886 ± 0.6	935.9 ± 0.6	Neutral Density Water Vapor
6	830 ± 2	445 ± 3	Surface Imaging, Diopter
7	802 ± 2	753 ± 2	Surface Imaging, Diopter
8	861 ± 2	753 ± 2	Surface Imaging
9	901 ± 2	753 ± 2	Left: Surface Imaging Right: Polarization
10	932 ± 2	604 ± 2	Surface Imaging
11	1001 ± 2	533 ± 2	Surface Imaging
12	967 ± 2	485 ± 2	Surface Imaging

Table 2.1.3 - SSI Spectral Filters (Engineering Model)

Filter Number	Left Eye Center Wavelength, nm	Right Eye Center Wavelength, nm	Application
1	672.9 ± 2	672.9 ± 2	Surface Stereo pair (RED)
2	446.6 ± 2	446.7 ± 2	Surface Stereo pair (BLUE)
3	450.8 ± 2	671.1 ± 0.5	Neutral Density Atmosphere
4	990.7 ± 0.6	935.5 ± 0.6	Neutral Density Water Vapor
5	870.0 ± 0.5	935.7 ± 2	Neutral Density Water Vapor
6	833.3 ± 2	449.6 ± 2	Surface Imaging, Diopter
7	801.5 ± 2	753.1 ± 2	Surface Imaging, Diopter
8	864.3 ± 2	754.2 ± 2	Surface Imaging
9	899.5 ± 2	753.5 ± 2	Left: Surface Imaging Right: Polarization
10	930.7 ± 2	603.8 ± 2	Surface Imaging
11	1002.0 ± 2	532.0 ± 2	Surface Imaging
12	968.5 ± 2	485.3 ± 2	Surface Imaging

2.2 Robotic Arm Camera (RAC)

The RAC is a RA-mounted camera with a spectral bandpass from approximately 450 nm (CCD cutoff) to 700 nm (short pass filter). It will primarily be used to plan trench digs performed by RA-mounted instruments and acquire image data of targets of interest for science analysis.

RAC optics characteristics useful in the analysis of EDR and RDR products are described in Table 2.2.1 below:

Table 2.2.1 - RAC Optics Characteristics

Characteristic	Value
Field of View (FOV)	26.5 deg x 13.3 deg, or 22 micron/pixel (close) 53.1 deg x 26.5 deg (far)
Angular Resolution	0.90 mrad/pixel at close focus (11 mm) 1.81 mrad/pixel at hyperfocus (28 cm to infinity)
Number of Spectral Filters	450 nm (CCD cutoff) – 700 nm (short pass filter)

The spectral bandwidths for the LEDs in the RAC are described in Table 2.2.2 below:

Table 2.2.2 - RAC Bandpasses

LED	Wavelength (Bandpass), nm	Application
Red	631 (21) at –35 deg C	Illumination, Color (when grouped with G, B)
Green	530 (40) at –35 deg C	Color (when grouped with R, B)
Blue	472 (30) at –35 deg C	Color (when grouped with R, G)

2.3 MECA Optical Microscope (MECA-OM)

The MECA-OM is a single fixed-focus 6:1 magnification (4 micrometer/pixel) camera in a dark enclosure that provides imaging primarily at a working distance of 13.9 microns. It will be used to analyze small grains of regolith (< 100 micrometers) from either RA excavations or airfall. It also provides context images for the Atomic Force Microscope (AFM). The CCD spectral bandpass cuts off at approximately 450 nm on the short wavelength end and above 1000 nm at the long end.

MECA-OM optics characteristics useful in the analysis of EDR and RDR products are described in Table 2.3 below:

Table 2.3 – MECA-OM Optics Characteristics

Characteristic	Value
Field of View (FOV)	2.04 mm (tall) x 1.02 mm (wide)
Image Size	512 (tall) x 256 (wide) in 12 bits
Depth of Field	> 50 µm (±25 µm)
Working Distance at Focus	13.9 µm
Pixel Scale	4 µm/pixel
Spectral Bandpass	450 nm – 1000 nm (limited by CCD response)

Except for fluorescence experiments, the practical spectral wavelength is determined by the LED illumination system, four LED's centered at 637 nm (red), 524 nm (green), 467 nm (blue) and 375 nm (UV). The UV LED is filtered such that the central wavelength does not drift appreciably with temperature. LED bandpasses are shown in Table 2.4 below:

Table 2.4 – MECA-OM Bandpasses

LED	Center of Bandpass, nm
Red	637
Green	524
Blue	467
UV	375

3. DATA PROCESSING OVERVIEW

3.1 Data Processing Level

This documentation uses the "Committee on Data Management and Computation" (CODMAC) data level numbering system. The PHX camera instrument EDRs referred to in this document are considered "Level 2" or "Edited Data" (equivalent to NASA Level 0). The EDRs are to be reconstructed from "Level 1" or "Raw Data", which are the telemetry packets within the project specific Standard Formatted Data Unit (SFDU) record. They are to be assembled into complete images, but will not be radiometrically or geometrically corrected.

PHX camera instrument RDRs are considered "Level 3" ("Calibrated Data" equivalent to NASA Level 1-A), "Level 4" ("Resampled Data" equivalent to NASA Level 1-B), or "Level 5" ("Derived Data" equivalent to NASA Level 1-C, 2 or 3). The RDRs are to be reconstructed from "Level 2" edited data, and are to be assembled into complete images that may include radiometric and/or geometric correction.

Refer to Table 3.1 for a breakdown of the CODMAC and NASA data processing levels.

Table 3.1 - Processing Levels for Science Data Sets

NASA	CODMAC	Description
Packet data	Raw - Level 1	Telemetry data stream as received at the ground station, with science and engineering data embedded.
Level 0	Edited - Level 2	Instrument science data (e.g., raw voltages, counts) at full resolution, time ordered, with duplicates and transmission errors removed.
Level 1-A	Calibrated - Level 3	Level 0 data that have been located in space and may have been transformed (e.g., calibrated, rearranged) in a reversible manner and packaged with needed ancillary and auxiliary data (e.g., radiances with the calibration equations applied).
Level 1-B	Resampled - Level 4	Irreversibly transformed (e.g., resampled, remapped, calibrated) values of the instrument measurements (e.g., radiances, magnetic field strength).
Level 1-C	Derived - Level 5	Level 1A or 1B data that have been resampled and mapped onto uniform space-time grids. The data are calibrated (i.e., radiometrically corrected) and may have additional corrections applied (e.g., terrain correction).
Level 2	Derived - Level 5	Geophysical parameters, generally derived from Level 1 data, and located in space and time commensurate with instrument location, pointing, and sampling.
Level 3	Derived - Level 5	Geophysical parameters mapped onto uniform space-time grids.

3.2 Data Generation

PHX camera instrument EDRs and RDRs will be generated by JPL's Multimission Instrument Processing Laboratory (MIPL) as part of the OPGS subsystem of the PHX GDS. RDRs will also be generated by the SSI, RAC and MECA-OM science instrument teams at the SOC facility at the University of Arizona, as well as at their home institutions (Texas A&M for SSI and RAC, JPL for MECA-OM).

3.2.1 EDR Data Product

As the fundamental image data archive product, the EDR will be generated as "raw" uncalibrated data within an automated pipeline process managed by MIPL under OPGS at JPL as part of the critical path in RA operations. The size of an EDR data product is approximately 2 MB.

3.2.1.1 Data Flow

The EDR processing begins with the reconstruction of packetized telemetry data resident on the TDS by MIPL's telemetry processor "phxtelemproc", which processes it with SPICE kernels provided by NAIF. The EDR will be generated within 60 seconds after the telemetered science instrument packets have been staged on the TDS. This data flow is illustrated in Figure 3.3.1.1.1, and is elaborated in the Data Format discussion:

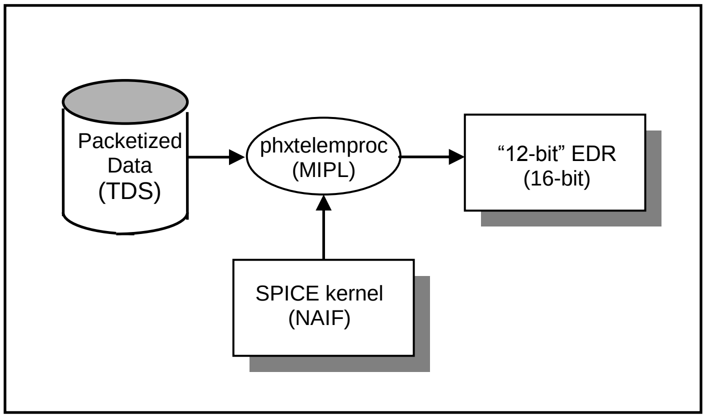


Figure 3.2.1.1 – EDR Generation

3.2.1.2 Data Format

The EDR will be formatted according to this SIS, such that each of the two radiometric formats of telemetry data will be stored "unscaled" in a signed 16-bit integer. In the first case, 12-bit data scaled onboard to 8-bit via a "12 to 8-bit" Lookup Table (LUT) or, via "shift-by-4" (SX4) bit shifting, will be downlinked as 8-bit data. During ground processing of the 8-bit data, as part of decompression, an Inverse Lookup Table (ILUT) will be applied to scale the lowest 8 bits to the lowest 12 bits in the signed 16-bit integer. In the second case, 12-bit data without onboard LUT scaling or bit shifting will be downlinked as 12-bit data and stored "as is" in the 12 lowest bits of the signed 16-bit integer.

There will not be multiple versions of the EDR. Missing packets will be identified and reported for retransmission to the ground as "partial datasets". Prior to retransmission, the missing EDR data will be filled with zeros. The EDR data will be reprocessed only after all "partial datasets" are retransmitted and received on the ground. In these cases, the original EDR version will be overwritten, retaining only a single version. The EDR data product will be placed into FEI for distribution to the Science Operations Center (SOC) at the University of Arizona and to facilitate the archiving process.

3.2.2 RDR Data Product

RDR data products will be generated by, but not limited to, MIPL using the Mars Suite of VICAR image processing software at JPL, the SSI and RAC science instrument teams using TAMCAL and RACCAL software at the SOC facility at the University of Arizona and at the teams' home institution at Texas A&M Univerisity, and the MECA Science Team using same set of SSI/RAC software tools at JPL. The RDRs produced will be "processed" data. The input will be one or more Camera EDR or RDR data products and the output will be formatted according to this SIS. Additional meta-data may be added by the software to the PDS label.

There may be multiple versions of a PHX Camera RDRs. The RDR data product will be placed into FEI for distribution.

3.3 Data Validation

Validation of the PHX EDRs will fall into two primary categories: automated and manual. Automated validation will be performed on every EDR product produced for the mission. Manual validation will only be performed on a subset.

Automated validation will be performed as a part of the archiving process and will be done simultaneously with the archive volume validation. Validation operations performed will include such things as verification that the checksum in the label matches a calculated checksum for the data product (i.e., that the data product included in the archive is identical to that produced by the real-time process), a validation of the PDS syntax of the label, a check of the label values against the database and against the index tables included on the archive volume, and checks for internal consistency of the label items. The latter include such things as verifying that the product creation date is later than the earth received time, and comparing the geometry pointing information with the specified target. As problems are discovered and/or new possibilities identified for automated verification, they will be added to the validation procedure.

Manual validation of the images will be performed both as spot-checking of data through-out the life of the mission, and comprehensive validation of a sub-set of the data (for example, a couple of days' worth of data). These products will be viewed by a human being. Validation in this case will include inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the target shown / apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure.

4. DATA PRODUCT OVERVIEW

The data in the EDR data product is a copy of the scene that had been projected onto the camera instrument's charge-coupled device (CCD) and shifted into the CCD memory buffer. That is, the EDR consists of unprocessed experiment data stored in binary format.

For SSI, a total of 1024 x 1024 image pixels plus 32 reference pixels per line are each digitized to 12 bits resolution. The binary data may be returned as 12-bit or 8-bit scaled data. The returned uncompressed 12–bit binary data is "packed" during transmission and stored in the EDR as a signed 16-bit integer. The returned uncompressed 8-bit binary data is also stored in the EDR as a signed 16-bit integer. Compressed data is JPEG encoded. The SSI camera acquisition of the scene and subsequent onboard storage and readout of image data is illustrated in Figure 4.1 below. Note that the Reference Pixels are returned separately from the rest of the image:

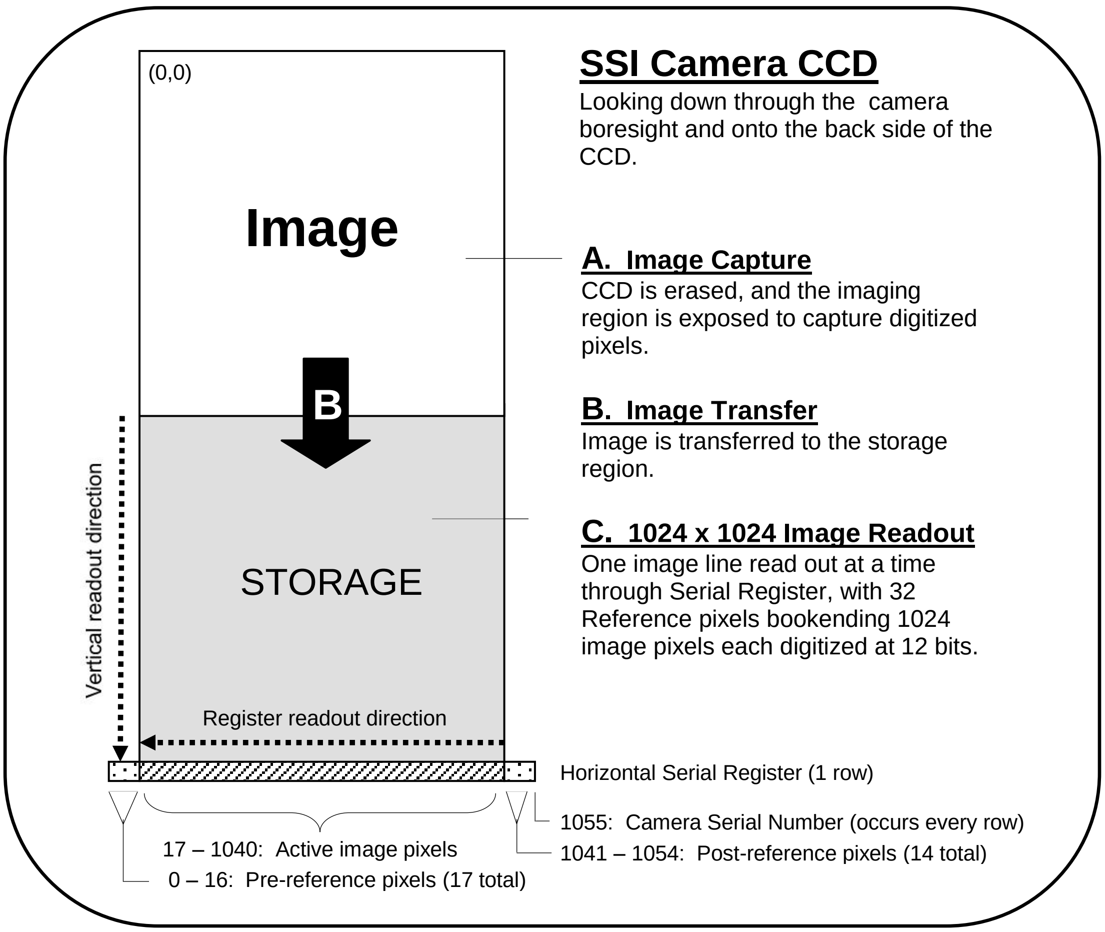


Figure 4.1 - Acquisition and Readout of SSI Image Data

For RAC, a total of 512 x 256 image pixels plus 8 null pixels and 8 dark pixels per line are each digitized to 12 bits resolution. As with SSI, the binary data may be returned as 12-bit or 8-bit scaled data. The returned uncompressed 12–bit binary data is "packed" during transmission and stored in the EDR as a signed 16-bit integer. The returned uncompressed 8-bit binary data is also stored in the EDR as a signed 16-bit integer. Compressed data is JPEG encoded. The camera acquisition of the scene and subsequent onboard storage and readout of image data is illustrated in Figure 4.2 below:

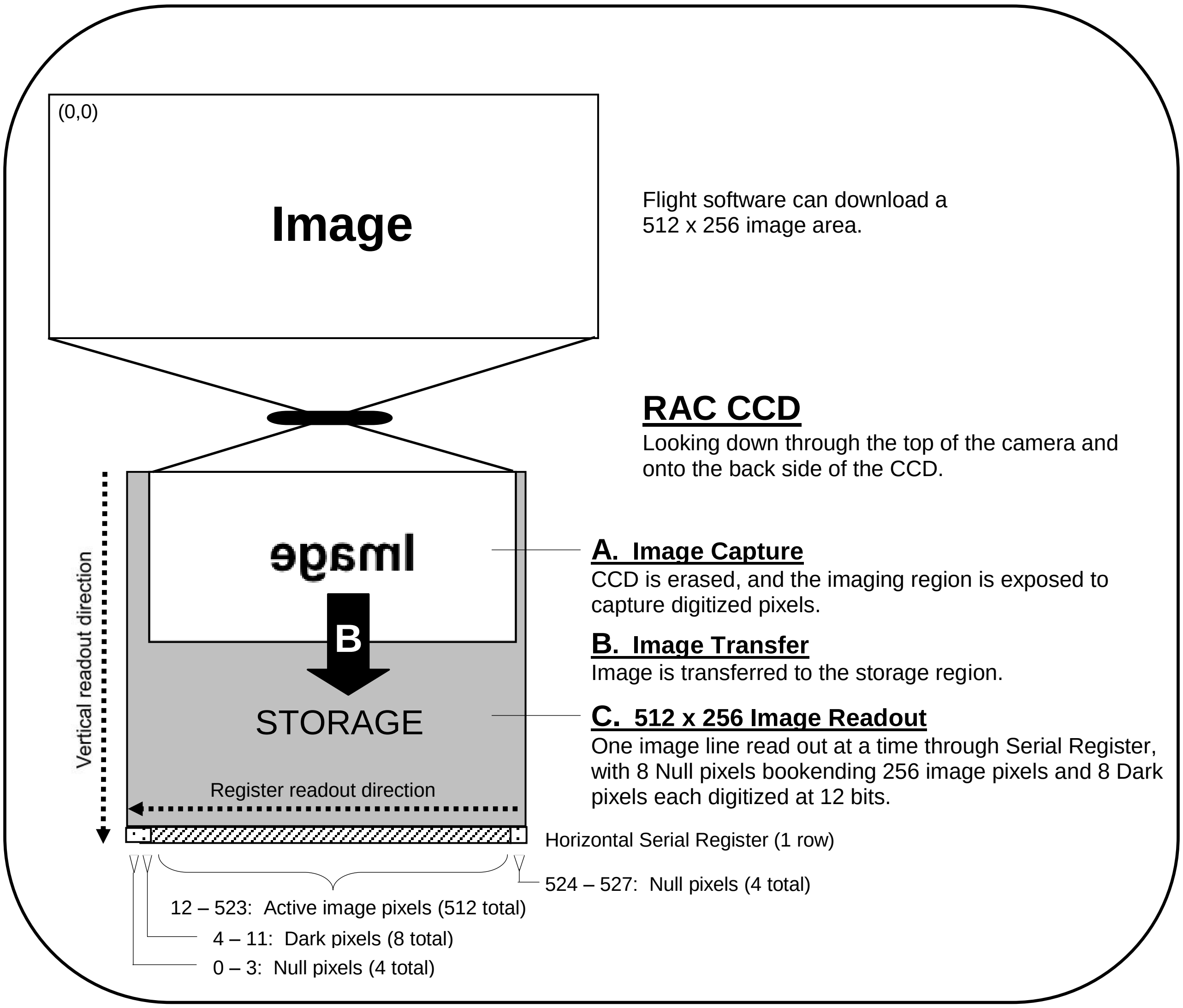


Figure 4.2 - Acquisition and Readout of RAC Image Data

For MECA-OM, a total of 512 x 256 image pixels plus 8 null pixels and 8 dark pixels per line are each digitized to 12 bits resolution. As with SSI and RAC, the binary data may be returned as 12-bit or 8-bit scaled data. The returned uncompressed 12–bit binary data is "packed" during transmission and stored in the EDR as a signed 16-bit integer. The returned uncompressed 8-bit binary data is also stored in the EDR as a signed 16-bit integer. Compressed data is JPEG encoded. The camera acquisition of the scene and subsequent onboard storage and readout of image data is illustrated in Figure 4.3 below:

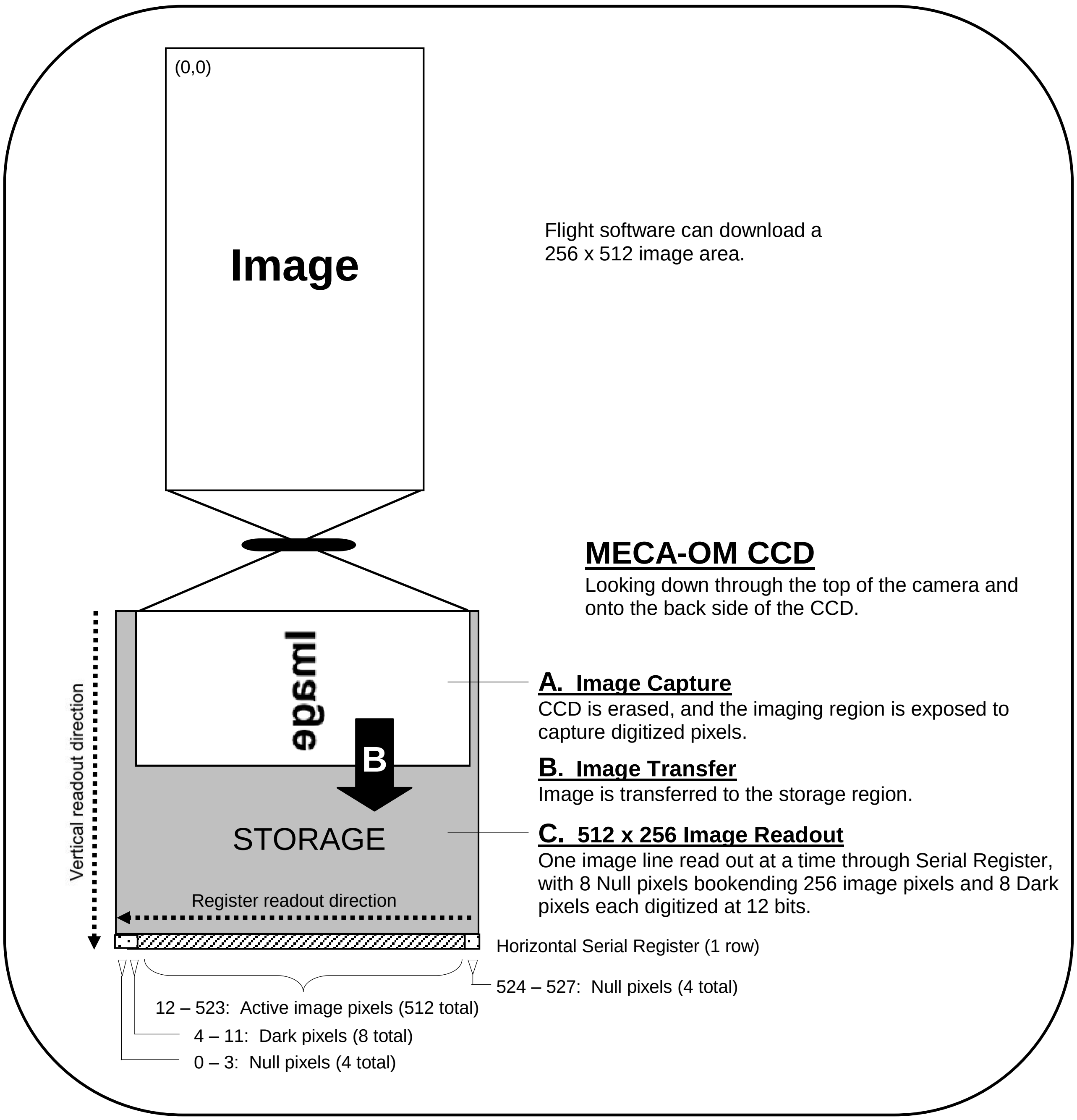


Figure 4.3 - Acquisition and Readout of MECA-OM Image Data

4.1 Data Product Structure

The EDR structure consists of an ASCII PDS label, followed by an embedded ASCII VICAR label, followed by a n x m block of binary image data with the origin at the upper left pixel in line (row) 1, sample (column) 1. Note that some camera EDR products will be rotated so that the origin (1,1) is not the same as the CCD origin.

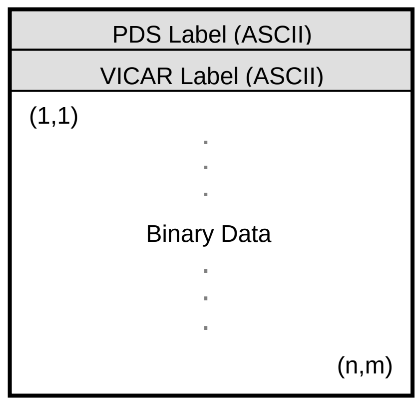


Figure 4.1.1 - EDR Structure

RDR products will have three possible structures. RDRs generated by MIPL will have a VICAR label wrapped by a PDS label, see Figure 4.1.2, Diagram A). RDR products not generated by MIPL may contain only a PDS label (Figure 4.1.2, Diagram B). Or, RDR products conforming to a standard other than PDS, such as JPEG compressed or certain Terrain products (Figure 4.1.2, Diagram C), are acceptable with a detached PDS header during mission operations. For a description of the PDS label, see Section 4.2.1, and for a description of the VICAR Label, see Section 4.2.2, and for a mapping between the two, see Section 4.2.3.

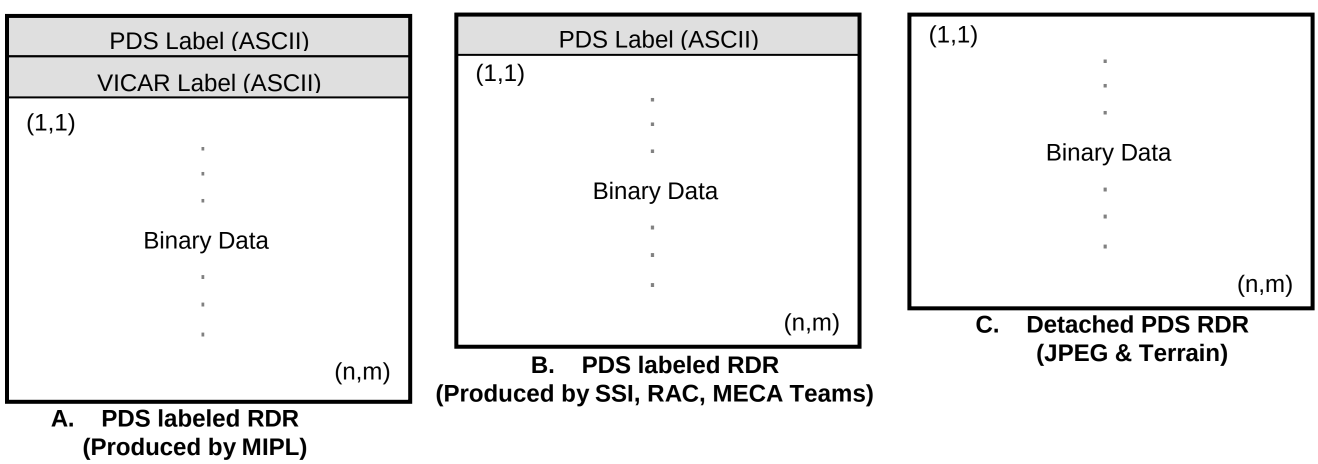


Figure 4.1.2 - RDR Structures

4.2 Label and Header Descriptions

4.2.1 PDS Label

PHX Camera Payload EDRs and RDRs, with the exception of the OPGS Terrain RDR, have an attached PDS label. The OPGS Terrain RDR has a detached PDS label. Each institution is responsible for converting PDS-formatted image products to be compatible with their own software systems (such as VICAR, IDL, ISIS, etc.).

A PDS label is object-oriented and describes the objects in the data file. The PDS label contains keywords for product identification. The label also contains descriptive information needed to interpret or process the data in the file.

PDS labels are written in Object Description Language (ODL) (see Reference 4). PDS label statements have the form of "keyword = value". Each label statement is terminated with a carriage return character (ASCII 13) and a line feed character (ASCII 10) sequence to allow the label to be read by many operating systems. Pointer statements with the following format are used to indicate the location of data objects in the file:

^object = location

where the carat character (^, also called a pointer) is followed by the name of the specific data object. The location is the 1-based starting record number for the data object within the file.

4.2.1.1 PDS Image Object

An IMAGE object is a two-dimensional array of values, all of the same type, each of which is referred to as a *sample*. IMAGE objects are normally processed with special display tools to produce a visual representation of the samples by assigning brightness levels or display colors to the values. An IMAGE consists of a series of lines, each containing the same number of samples.

The required IMAGE keywords define the parameters for simple IMAGE objects:

- LINES is the number of lines in the image.
- LINE_SAMPLES is the number of samples in each line.
- SAMPLE_BITS is the number of bits in each individual sample.
- SAMPLE_TYPE defines the sample data type.

The IMAGE object has a number of keywords relating to image statistics. These keywords will be present in all EDRs. In RDRs, they are optional, and if they are present, they must be updated to reflect the current statistics of the image (often they will be omitted for the sake of computational efficiency). Note that the VICAR label never contains these keywords; see section 4.2.3. The statistics keywords are:

- MEAN
- MEDIAN
- MAXIMUM
- MINIMUM
- STANDARD_DEVIATION
- CHECKSUM

Many variations on the basic IMAGE object are possible with the addition of optional keywords and/or objects. The "^IMAGE" keyword identifies the start of the image data and will skip over the VICAR label. Recommended image formats are described and illustrated in Reference 4, Appendix A.19.

4.2.1.2 Keyword Length Limits

All PDS keywords are limited to 30 characters in length (Section 12.7.3 in PDS Standards Reference). Therefore, software that reads PHX PDS labels must be able to ingest keywords up to 30 characters in length.

For RDR producing institutions wishing to accommodate the VICAR mapping (see Section 4.2.3) of PDS keywords that use a <*unit*> tag after the value, such keywords must be limited to 26 characters in length. Otherwise, those keywords will not be transcoded from the PDS label into a VICAR label.

4.2.1.3 Data Type Restrictions

In order to accommodate VICAR dual-labeled files, 16-bit data must be stored as signed data. Unsigned 16-bit data is not supported. 12-bit unsigned data from the cameras is stored in a 16-bit signed value.

4.2.1.4 Interpretation of N/A, UNK, and NULL

During the completion of data product labels or catalog files, one or more values may not be available for some set of required data elements. In this case PDS provides the symbolic literals "N/A", "UNK", and "NULL", each of which is appropriate under different circumstances.
As a note, if any one of these three symbolic literals are used in place of a keyword value that is normally followed by a Unit Tag(s) (e.g., "<*value*>"), the Unit Tag(s) is <u>removed</u> from the label.

- "N/A" ("Not Applicable") indicates that the values within the domain of this data element are not applicable in this instance. For example, a data set catalog file describing NAIF SPK kernels would contain the line:

  INSTRUMENT_ID = "N/A"

  because this data set is not associated with a particular instrument.

  "N/A" may be used as needed for data elements of any type (e.g., text, date, numeric, etc.).

- "UNK" ("Unknown") indicates that the value for the data element is not known and never will be. For example, in a data set comprising a series of images, each taken with a different filter, one of the labels might contain the line:

  FILTER_NAME = "UNK"

  if the observing log recording the filter name was lost or destroyed and the name of the filter is not otherwise recoverable.

  "UNK" may be used as needed for data elements of any type.

- "NULL" is used to flag values that are *temporarily* unknown. It indicates that the data preparer recognizes that a specific value should be applied, but that the true value was not readily available. "NULL" is a placeholder. For example, the line:

DATA_SET_RELEASE_DATE = "NULL"

might be used in a data set catalog file during the development and review process to indicate that the release date has not yet been determined.

“NULL” may be used as needed for data elements of any type.

Note that all “NULL” indicators should be replaced by their actual values prior to final archiving of the associated data.

4.2.1.5 PDS Label Constructs “Class”, “Object” and “Group”

The PDS has designed a set of formal and informal constructs for labeling data products. In the PDS realm, “formal” infers a standardized design or set of rules that provides a protocol across multiple data products (e.g., multiple flight missions) for PDS validation tools, and involves a rigorous approval process. “Informal” infers a less rigorous process by which the construct meets PDS approval. For both formal and informal constructs, the member keywords must be defined in the *Planetary Science Data Dictionary* (PSDD) [Ref 18]. In PHX Camera EDRs and RDRs, the PDS Label includes the following “formal” and “informal” constructs:

- *Class* - The Class construct is informal and resides in a PDS label as a grouping of keywords that are thematically tied together. Classes are usually preceded by a label comment, although it is not required. PDS label comments are character strings bounded by “/* */” characters.

  In the PHX Camera PDS label a Class of keywords will be preceded by a comment string as follows:

  ```
  /*  comment string  */
  keyword      =   keyword value
  keyword      =   keyword value
  ```

- *Object* - The Object construct is formal and is a set of standard keywords used for a particular data product. In the PSDD, each Object definition lists the elements required to be present each time the Object is used in a product label. The PSDD also provides a list of additional, optional keywords that are frequently used in the Object. Any element defined in the PSDD may be included as an optional element in any Object definition, at the discretion of the data preparer.

  In the PHX Camera PDS label an Object’s set of keywords is specified as follows:

  ```
  OBJECT          =   Object identifier
      keyword     =   keyword value
      keyword     =   keyword value
  END_OBJECT      =   Object identifier
  ```

- *Group* - The Group construct can be either a formal or informal grouping of keywords that are not components of a larger Object. Group keywords may reside in more than one Group within the label.

The Group construct is further described in section 12.4.5 of the PDS Standards Reference, "Object Description Language Specification and Usage: GROUP Statement".

In the PHX Camera PDS label, a Group's set of keywords is specified as follows:

```
GROUP          =  Group identifier
    keyword    =  keyword value
    keyword    =  keyword value
END_GROUP      =  Group identifier
```

4.2.2 VICAR Label

For all EDR data products and MIPL produced RDR data products, an embedded VICAR label follows the PDS label and is pointed to by the PDS pointer "^IMAGE_HEADER". The VICAR label is also organized in an ASCII, "keyword = value" format, although there are only spaces between keywords (no carriage return/line feeds as in PDS). The information in the VICAR label is an exact copy of the information in the PDS label as defined in the next section. The reader is referred to the VICAR File Format document for details of the format, which is available at the URL "http://www-mipl.jpl.nasa.gov/vicar/vic_file_fmt.html". The following text is an excerpt which describes the basic structure:

> A VICAR file consists of two major parts: the labels, which describe what the file is, and the image area, which contains the actual image. The labels are potentially split into two parts, one at the beginning of the file, and one at the end. Normally, only the labels at the front of the file will be present. However, of the EOL keyword in the system label (described below) is equal to 1, then the EOL labels (End Of file Labels) are present. This happens if the labels expand beyond the space allocated for them. The VICAR file is treated as a series of fixed-length records, of size RECSIZE (see below). The image area always starts at a record boundary, so there may be unused space at the end of the label, before the actual image data starts.
>
> The label consists of a sequence of "keyword=value" pairs that describe the image, and is made up entirely of ASCII characters. Each keyword-value pair is separated by spaces. Keywords are strings, up to 32 characters in length, and consist of uppercase characters, underscores ("_"), and numbers (but should start with a letter). Values may be integer, real, or strings, and may be multiple (e.g. an array of 5 integers, but types cannot be mixed in a single value). Spaces may appear on either side of the equals character (=), but are not normally present. The first keyword is always LBLSIZE, which specifies the size of the label area in bytes. LBLSIZE is always a multiple of RECSIZE, even if the labels don't fill up the record. If the labels end before LBLSIZE is reached (the normal case), then a 0 byte terminates the label string. If the labels are exactly LBLSIZE bytes long, a null terminator is not necessarily present. The size of the label string is determined by the occurrence of the first 0 byte, or LBLSIZE bytes, whichever is smaller. If the system keyword EOL has the value 1, then End-Of-file Labels exist at the end of the image area (see above). The EOL labels, if present, start with another LBLSIZE keyword, which is treated exactly the same as the main LBLSIZE keyword. The length of the EOL labels is the smaller of the length to the first 0 byte or the EOL's LBLSIZE. Note that the main LBLSIZE does not include the size of the EOL labels. In order to read in the full label string, simply read in the EOL labels, strip off the LBLSIZE keyword, and append the rest to the end of the main label string.

4.2.3 Mapping of PDS and VICAR Labels

The information contained in the PDS and VICAR embedded labels are identical, by definition. Either label may be used interchangeably, for any purpose in the mission. Any MIPL software that modifies one label must also modify the other. This is often most easily accomplished by stripping off one of the headers, processing the remaining label as desired locally, and then running a conversion tool to re-create the missing header. Such tools will be provided by MIPL.

It is important to note that these files are simultaneously valid PDS images, and valid VICAR images, and may be processed equally by tools of either system. It is critical for the integrity of the data that both labels be maintained, as described above.

The mapping between PDS keywords is straightforward. Appendix A shows a label in PDS format. For space reasons, the corresponding VICAR label is omitted from this document, but it is required. The mapping rules are as follows:

- Keyword values are identical in both cases. The only changes to values are those mandated by the file format itself, such as quoting rules. See the respective PDS and VICAR documents for details, but in general, PDS uses double quotes (") while VICAR uses single quotes (').

- With the exception of keywords defining the file format itself (described below), keyword names are identical in both cases.

- Any PDS group maps 1-to-1 to a VICAR property set with the same name (group name == property set name). All contained keywords are identical in both cases. The GROUP and END-GROUP keywords are omitted from the VICAR label; PROPERTY keywords are used instead (as per the VICAR file format definition).

- Any set of PDS keywords not in a group (in PDS terms, a class) is identified by an introductory comment (e.g. /* IDENTIFICATION DATA ELEMENTS */). Such classes map 1-to-1 to a VICAR property set. The name of the VICAR property set and the name of the PDS introductory comment map as follows:

Table 4.2.3 - PDS Class to VICAR Property Set Mappings

PDS Class Comment	VICAR Property Set Name
/* FILE DATA ELEMENTS */	special case, see below
/* POINTERS TO DATA OBJECTS */	special case, see below
/* IDENTIFICATION DATA ELEMENTS */	IDENTIFICATION
/* TELEMETRY DATA ELEMENTS */	TELEMETRY
/* HISTORY DATA ELEMENTS */	PDS_HISTORY
/* COMPRESSION RESULTS */	COMPRESSION_PARMS

- PDS comments (i.e., /* *string* */) are stored in a VICAR keyword named "PDS_COMMENT". This keyword appears in the VICAR property containing the elements immediately following the comment. When converting from VICAR to PDS, the comment is placed immediately before the group or class. Blank lines should surround the comment. Note that with OPGS-generated EDR and RDR data products, multiple comment lines in a Group are not supported.

- The PDS objects IMAGE_HEADER and IMAGE, as well as the keywords in /* FILE DATA ELEMENTS */ and /* POINTERS TO DATA OBJECTS */ in the table above, do not map directly to VICAR. They all describe the layout of the file and the image data. The VICAR equivalent for all of these items is the VICAR System label. Information maps between these in a straightforward way. It should be trivial to construct a VICAR system label and the above-referenced PDS entities after referring to the respective file-format-definition documents. Note that the /* FILE DATA ELEMENTS */ and /* POINTERS TO DATA OBJECTS */ comments are constant and so are not mapped to PDS_COMMENT keywords in the VICAR label. They are inserted automatically as part of the system label conversion process.

- The statistics-related keywords in the PDS IMAGE object are MEAN, MEDIAN, MAXIMUM, MINIMUM, STANDARD_DEVIATION, and CHECKSUM. These keywords are never transferred to the VICAR label. For VICAR -> PDS conversion, they can be computed from the image, or simply omitted from the PDS image (for RDRs only - EDRs require them).

- A few remaining items in the PDS_IMAGE object are treated specially. The FIRST_LINE, FIRST_LINE_SAMPLE, INVALID_CONSTANT, and MISSING_CONSTANT keywords are tranferred to the VICAR IMAGE_DATA property set.

- Any PDS keyword with a <*unit*> tag after the value is transferred to the VICAR label without the unit tag. A VICAR keyword with the same name, but with "__UNIT" (two underscores) appended to the end, is added with the value of the unit. So for example, the PDS keyword "EXPOSURE_TIME = 1.5 <s>" would translate to two VICAR keywords: "EXPOSURE_TIME = 1.5" and "EXPOSURE_TIME__UNIT = SEC". Note that because of this, any PDS keyword that can support a unit is limited to 26 characters. If there is more than one value (an array), a unit is associated with each. In this case, the "__UNIT" VICAR keyword becomes multi-valued also, with each unit copied in sequence. If one of the elements does not have a unit (but others do), the corresponding entry is "N/A" (which is not copied to the PDS label). So for example, PDS "CONTRIVED_ANGLE = (1.2 <rad>, 22.0, 54.1 <deg>)" would map to VICAR "CONTRIVED_ANGLE = (1.2, 22.0, 54.1)" and "CONTRIVED_ANGLE__UNIT = (RAD, N/A, DEG)".

- The VICAR history label is omitted from the PDS header

4.3 Binary Data Storage Conventions

PHX camera instrument EDR data are stored as binary data. The data are 12-bit integers stored in signed 16-bit integers, or rescaled 8-bit integers stored in signed 16-bit integers with only the lowest ordered 8 bits being valid. The PDS and VICAR labels are stored as ASCII text.

4.3.1 Bit and Byte Ordering

The ordering of bits and bytes is only significant for pixel data; all other labeling information is in ASCII.

For non-byte data, which includes 16-bit signed shorts, 32-bit signed ints, and 32- and 64-bit IEEE floating-point numbers, the data may be stored in either Most Significant Byte first ("big-endian", as used by e.g. Sun computers and Java), or Least Significant Byte first ("little-endian", as used by e.g. Linux and Windows computers). This follows both the PDS and VICAR file format conventions.

For PDS, the SAMPLE_TYPE label in the IMAGE object defines which ordering is used in the file. For VICAR, the INTFMT and REALFMT labels in the System label define the ordering. See the respective PDS and VICAR file format definition documents.

Both file formats specify that bit 0 is the least significant bit of a byte. PHX EDR's may be constrained to use MSB only, but RDR's still need to be flexible.

Table 4.3.1 - PHX Camera EDR and RDR Bit Ordering

Address	MSB-first	LSB-first
n	most significant byte	least significant byte
n+1	next	next
n+2	next	next
n+3	least significant byte	most significant byte

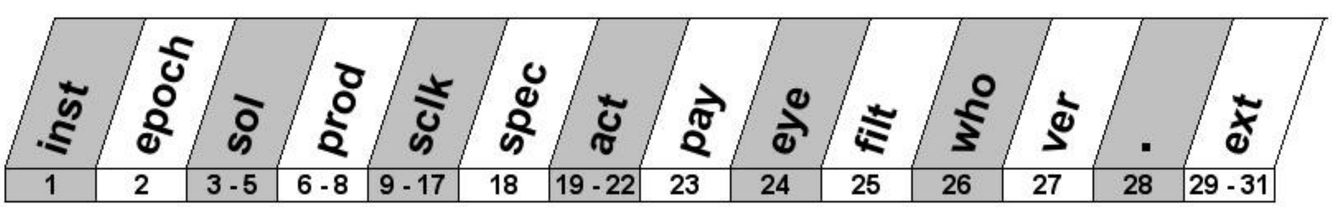
4.4 File Naming

There are three file naming schemes adapted for the PHX image and non-image data products. The first applies to the EDR data product and all Single-frame RDR data products. The second applies to all Mosaic RDR data products. Both file naming schemes adhere to the Level II 27.3 filename convention to be compliant with PDS standards. The third applies to Terrain products, and does not adhere to the PDS Level II 27.3 filename standard.

4.4.1 EDR and Single-frame RDR

Each PHX EDR and Single-frame RDR data product can be uniquely identified by incorporating into the product filename the Instrument identifier, the Mission source/epoch, the Mars solar day (Sol), the data Product Type, the Starting Spacecraft Clock count (SCLK) of the camera event, the Activity ID as extracted from the round-trip accountability Token, the Payload identifier from the Token, a flag denoting Special processing, the camera “Eye”, the spectral Filter, the product Creator identifier and a Version number. For all EDR products, the Creator field will display the character “M” (for **M**IPL). For RDR products, the Creator field will display any single character from a variety of characters.

The Single-frame RDR data products that share the naming scheme with the EDR data product are numerous. They are listed in the description of the Product Type field found in the filename convention definition, which follows:



where,

inst = (1 alpha character) PHX science instrument identifier.

Valid values for PHX camera instruments:

"**S**" - SSI "**O**" - MECA-OM
"**R**" - RAC

Valid values for PHX instruments not described in this SIS:

"**A**" - RA "**M**" - MET-P&T
"**D**" - MARDI ("D" for Descent) "**P**" - MECA-TECP
"**E**" - ESE "**T**" - TEGA
"**F**" - MECA-AFM "**W**" - MECA-WCE
"**L**" - MET-LIDAR "**X**" - MECA-Misc

epoch = (1 alpha character) PHX Mission source/epoch. Valid values:

"**S**" - Surface, flight model "**C**" - Cruise, flight model
"**T**" - Testbed

sol = (3 integers) If Epoch is "S", specifies number of Solar days since first full day on Mars. Landing day is Sol one ("001"). However, if Epoch is "T" or "C", specifies day-of-year (ERT or SCET). Example value is "004".

prod = (3 alpha characters) Product Type identifier of input data. Product types are differentiated as having camera-induced distortion removed ("linearized") or not removed (nominal). Four special flag characters follow:

a) Beginning "**E**" – Type of **E**DR, which are raw with no camera model "linearization"or radiometric correction. If no beginning "E", then it is an RDR.

b) Ending "**L**" – If no beginning "E", denotes an RDR that is "**L**inearized".

Valid values for PHX camera instrument input data products:

Data Product	Non-linearized (NOMINAL)	Linearized
Full frame EDR	"**EFF**"	n/a
Sub-frame EDR	"**ESF**"	n/a
Downsampled EDR	"**EDN**"	n/a
Dark Current EDR	"**EDK**"	n/a
Reference Pixels EDR	"**ERP**"	n/a
Geometrically-corrected (linearized) Full frame	n/a	"**FFL**"
Geometrically-corrected (linearized) Sub-frame	n/a	"**SFL**"
Geometrically-corrected (linearized) Downsampled	n/a	"**DNL**"
Radiometrically-corrected RDR calibrated to absolute radiance units	"**RAD**"	"**RAL**"
Radiometrically-corrected RDR contrast enhanced	"**RSD**"	"**RSL**"
Radiometrically-corrected RDR calibrated to I/F radiance factor	"**IOF**"	"**IOL**"
Disparity RDR	"**DIS**"	"**DIL**"
Disparity of Samples RDR	"**DSS**"	"**DSL**"
Disparity of Lines RDR	"**DLS**"	"**DLL**"
XYZ RDR	"**XYZ**"	"**XYL**"
XYZ Lander Vol Exclusion Mask RDR	"**MSK**"	"**MSL**"
Masked XYZ RDR	"**XMZ**"	"**XML**"
X Component RDR	"**XXX**"	"**XXL**"
Y Component RDR	"**YYY**"	"**YYL**"
Z Component RDR	"**ZZZ**"	"**ZZL**"
Range (Distance) RDR	"**RNG**"	"**RNL**"

UVW (XYZ) Surface Normal RDR	"**UVW**"	"**UVL**"
UVW (XYZ) Surface Normal RDR	"**UVW**"	"**UVL**"
U (X) Surface Normal RDR	"**UUU**"	"**UUL**"
V (Y) Surface Normal RDR	"**VVV**"	"**VVL**"
W (Z) Surface Normal RDR	"**WWW**"	"**WWL**"
Surface Roughness RDR	"**RUF**"	"**RUL**"
Slope RDR	"**SLP**"	"**SLL**"
Slope Heading RDR	"**SHP**"	"**SHL**"
Slope Magnitude RDR	"**SMP**"	"**SML**"
RA Reachability RDR	"**IDD**"	"**IDL**"

sclk = (9 integers) Starting Spacecraft Clock time.

spec = (1 alphanumeric) Special processing designator to be defined later. This character is used to indicate off-nominal or special processing of the image. Examples might be use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.

The meaning of any individual character in this field will be defined on an ad-hoc basis as needed during the mission. Within one Activity ID, the character will be used consistently, so this field can be used to group together all derived products resulting in one kind of special processing. An attempt will be made to maintain consistency across different Activity ID's as well, but this may not always be possible; thus the meaning of characters may change across different Activity ID's.

A text file will be maintained containing all special processing designators that are used, the Activity ID's they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.

Valid values are:

"_" - Nominal processing

"<*value*>" - Arbitrary value to be defined later that flags a specific type of Special processing

act = (4 alphanumeric) Activity ID as extracted from round-trip accountability Token. Valid values are assigned at planning time by PSI tool.

pay = (1 alphanumeric) Identifier of Payload type as extracted from round-trip accountability Token. 1 Hex character representing 4 bits of the 32-bit Token based on a bit lookup, which follows:

Instrument	Bit String	Token Flag (<*observation type*>)	Valid Value
SSI, RAC	0000	Non-mosaic, Non-stereo (general science)	"**0**"
	0001	Non-mosaic, Non-stereo (calibration)	"**1**"
	0010	Non-mosaic, Non-stereo, Mesh/Reach (operations)	"**2**"
	0011	Non-mosaic, Non-stereo (atmospheric)	"**3**"
	0100	Non-mosaic, Stereo (general science)	"**4**"
	0101	Non-mosaic, Stereo (calibration)	"**5**"
	0110	Non-mosaic, Stereo, Mesh/Reach (operations)	"**6**"
	0111	Non-mosaic, Stereo (atmospheric)	"**7**"
	1000	Mosaic, Non-stereo (general science)	"**8**"
	1001	Mosaic, Non-stereo (calibration)	"**9**"
	1010	Mosaic, Non-stereo, Mesh/Reach (operations)	"**A**"
	1011	Mosaic, Non-stereo (atmospheric)	"**B**"

	1100	Mosaic, Stereo (general science)	"**C**"
	1101	Mosaic, Stereo (calibration)	"**D**"
	1110	Mosaic, Stereo, Mesh/Reach (operations)	"**E**"
	1111	Mosaic, Stereo (atmospheric)	"**F**"
MECA-OM	0000	Single, Single focus, monochrome (Clean)	"**0**"
	0001	Single, Single focus, monochrome (Exposed)	"**1**"
	0010	Single, Single focus, color (Clean)	"**2**"
	0011	Single, Single focus, color (Exposed)	"**3**"
	0100	Single, Through focus, monochrome (Clean)	"**4**"
	0101	Single, Through focus, monochrome (Exposed)	"**5**"
	0110	Single, Through focus, color (Clean)	"**6**"
	0111	Single, Through focus, color (Exposed)	"**7**"
	1000	Mosaic, Single focus, monochrome (Clean)	"**8**"
	1001	Mosaic, Single focus, monochrome (Exposed)	"**9**"
	1010	Mosaic, Single focus, color (Clean)	"**A**"
	1011	Mosaic, Single focus, color (Exposed)	"**B**"
	1100	Mosaic, Through focus, monochrome (Clean)	"**C**"
	1101	Mosaic, Through focus, monochrome (Exposed)	"**D**"
	1110	Mosaic, Through focus, color (Clean)	"**E**"
	1111	Mosaic, Through focus, color (Exposed)	"**F**"

eye = (1 alpha character) Camera eye. Valid values are:

"**L**" - Left camera eye
"**R**" - Right camera eye
"**M**" - Monoscopic (non-stereo camera)
"**A**" - 3-banded Anaglyph of Left, Right, Right eyes mapped to Red, Green, Blue channels
"**N**" - Not Applicable

filt = (1 alphanumeric) Spectral filter position or LED status.

Valid values:

Instrument	Filter Number	LED Status
SSI	Hex	n/a
RAC	n/a	"**R**" = Red
	n/a	"**G**" = Green
	n/a	"**B**" = Blue
	n/a	"**D**" = Dark/off
	n/a	"**O**" = other (combination of multiple LED's)
MECA-OM	n/a	"**R**" = Red
	n/a	"**G**" = Green
	n/a	"**B**" = Blue
	n/a	"**D**" = Dark/off
	n/a	"**W**" = White
	n/a	"**U**" = UV
	n/a	"**O**" = other (combination of multiple LED's)

who = (1 alpha character) Product Creator indicator. Valid values are:

"**A**" - University of **A**rizona
"**O**" - MECA-**O**M Team

“**M**” - **M**IPL (OPGS) at JPL
“**T**” - **T**exas A&M (Mark Lemmon)
“**X**” - Other

ver = (1 alphanumeric) Version identifier providing uniqueness for book keeping.

The valid values, in their progression, are as follows:

Range 1 thru 9 - **“1”, “2”… “9”**
Range 10 thru 35 - **“A”, “B” …“Z”**

The Version number increments by one whenever an otherwise-identical filename would be produced, independent of the Special field. Thus, even though the Special field may change, the Version number continues to increment by one over the previous Version. This allows the "best" Version of a product to be determined - irrespective of what special processing was done to achieve it - by simply looking for the highest Version number.

The following examples show how the Version field increments independently of the Special field thru a progression of RDR processing for the same SCLK product:

Examples		
Iteration of Processing	Special field value	Version field value
1	“_”	“1”
2	“B”	“2”
3	“B”	“3”
4	“E”	“4”
5	“_”	“5”
6	“A”	“6” (best)

ext = (2 to 3 alpha characters) Product type extension.

Valid values for nominal operations camera data products:

“**IMG**” - Camera image EDRs and RDRs (PDS labeled).
“**VIC** ” - Camera image EDRs and RDRs with only VICAR label (no PDS label)

Valid values for quick-look JPEG compressed camera data products:

“**JPG**” - JPEG compressed (no PDS label).

Valid values for In-situ instrument data products <u>not</u> described in this SIS:

“**QUB**” - Multi-layer spectral cube data
“**DAT**” - Non-imaging instrument data
“**LBL**” - Detached PDS labels for non-imaging data
“**TAB**” - tabularized data

Examples:

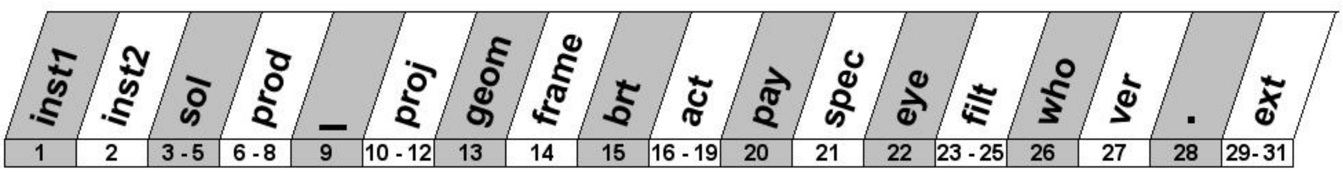
a) RS004EFF123456789_002C3M0M1.IMG — RAC instrument ("R"), Surface mission phase ("S"), Sol 4 (“004”), Full Frame EDR (“EFF”), SCLK 123456789 ("123456789"), Nominal processing (“_”), Activity ID “002C” (“002C”), Non-mosaic/Non-stereo (“3”), Monoscopic Eye ("M"), Filter position 0 ("0"), produced by MIPL ("M"), product version 1 ("1"), PDS-labeled ("IMG")

b) SS023RAD123456789U00C4CL2T1.IMG — SSI instrument ("S"), Surface mission phase ("S"), Sol 23 (“023”), non-linearized Rad-corrected RDR (“RAD”), SCLK 123456789 ("123456789"), flag “U” as an arbitrary type of Special processing determined by the product's Provider ("U"), Activity ID "00C4" ("00C4"), Mosaic/Stereo ("C"), Left Eye ("L"), Filter position 2 ("2"), produced by Texas A&M ("T"), product version 1 ("1"), PDS-labeled ("IMG")

4.4.2 Mosaic RDR

The PHX camera Mosaic RDR data products are usually derived from multiple EDR or RDR data products mosaicked together, although they can also be derived from single data products. They are uniquely identified by incorporating into the product filename the Lander mission identifier, the "primary" Instrument identifier, the "secondary" Instrument identifier, the Starting Sol denoting the start of mosaic data, the geometric Projection type, the Product Type ingested to build the mosaic, the Starting Site location, the Lander's Starting Position within the site, the camera "Eye", the spectral Filter, the product Creator identifier and a Version number.

The filename convention follows:



where,

inst1 = (1 alpha character) PHX science instrument identifier of "primary" instrument.

Valid values for PHX camera instruments:

"**S**" - SSI
"**O**" - MECA-OM
"**R**" - RAC

inst2 = (1 alpha character) PHX science instrument identifier of "secondary" instrument.

Valid values for PHX camera instruments:

"**S**" - SSI
"**O**" - MECA-OM
"**R**" - RAC
"_" - denotes NO secondary instrument

sol = (3 integers) Specifies number of Solar days since first full day on Mars. Landing day is Sol one ("001"). This field is defined as the primary Sol for this mosaic. For most (and all automatically-produced) products, this is extracted from the input image with the *earliest* (i.e. lowest) SCLK.

prod = (3 alpha characters) Product Type identifier of input data. Product types are differentiated as having camera-induced distortion removed ("linearized") or not removed (nominal). Four special flag characters follow:

a) Beginning "**E**" – Type of **E**DR, which are raw with <u>no</u> camera model "linearization"or radiometric correction. If no beginning "E", then it is an RDR.
b) Ending "**L**" – If no beginning "E", denotes an RDR that is "**L**inearized".

Valid values for PHX camera instrument input data products:

Data Product	Non-linearized (NOMINAL)	Linearized

Full frame EDR	"**EFF**"	n/a
Sub-frame EDR	"**ESF**"	n/a
Downsampled EDR	"**EDN**"	n/a
Dark Current EDR	"**EDK**"	n/a
Reference Pixels EDR	"**ERP**"	n/a
Geometrically-corrected (linearized) Full frame RDR	n/a	"**FFL**"
Geometrically-corrected (linearized) Sub-frame RDR	n/a	"**SFL**"
Geometrically-corrected (linearized) Downsampled RDR	n/a	"**DNL**"
Radiometrically-corrected RDR calibrated to absolute radiance units	"**RAD**"	"**RAL**"
Radiometrically-corrected RDR contrast enhanced	"**RSD**"	"**RSL**"
Radiometrically-corrected RDR calibrated to I/F radiance factor	"**IOF**"	"**IOL**"
Disparity RDR	"**DIS**"	"**DIL**"
Disparity of Samples RDR	"**DSS**"	"**DSL**"
Disparity of Lines RDR	"**DLS**"	"**DLL**"
XYZ RDR	"**XYZ**"	"**XYL**"
XYZ Lander Vol Exclusion Mask RDR	"**MSK**"	"**MSL**"
Masked XYZ RDR	"**XMZ**"	"**XML**"
X Component RDR	"**XXX**"	"**XXL**"
Y Component RDR	"**YYY**"	"**YYL**"
Z Component RDR	"**ZZZ**"	"**ZZL**"
Range (Distance) RDR	"**RNG**"	"**RNL**"
UVW (XYZ) Surface Normal RDR	"**UVW**"	"**UVL**"
U (X) Surface Normal RDR	"**UUU**"	"**UUL**"
V (Y) Surface Normal RDR	"**VVV**"	"**VVL**"
W (Z) Surface Normal RDR	"**WWW**"	"**WWL**"
Surface Roughness RDR	"**RUF**"	"**RUL**"
Slope RDR	"**SLP**"	"**SLL**"
Slope Heading RDR	"**SHP**"	"**SHL**"
Slope Magnitude RDR	"**SMP**"	"**SML**"

proj = (3 alpha characters) Projection type. Indicates the projection or perspective of the product. Valid values are:

- "**CYL**" - Cylindrical projection
- "**PER**" - Camera Point Perspective
- "**CYP**" - Cylindrical-Perspective projection
- "**POL**" - Polar projection
- "**VRT**" - Vertical projection
- "**ORT**" - Orthographic projection

geom = (1 alpha character) Specifies Geometry Correction flag. Indicates what kind of geometric correction has been applied to data. If multiple types of corrections were applied, then the highest-level (last in the table) value is used.

Valid values are:

- "_" - No correction (raw pointing)
- "**A**" - Auto-correction via tiepointing
- "**F**" - Auto-correction via tiepointing & auto-registration w/ fiducials
- "**G**" - Auto-correction via tiepointing & manual registration w/ fiducials
- "**T**" - Manual tiepointing
- "**R**" - Manual tiepointing & auto-registration w/ fiducials
- "**M**" - Manual tiepointing & manual registration w/ fiducials
- "**O**" - Other correction not listed above

frame = (1 alpha character) Specifies coordinate frame in which the mosaic is generated. Note that the instance of frames (e.g. Payload) is not specified and must be obtained from label.

Valid values are:

- "**S**" - Site frame (+X or 0 deg azimuth points North, +Z or nadir points down along local gravity vector)
- "**L**" - Local Level frame (effectively the same as Site frame, although translation offset is possible)
- "**P**" - Payload frame (+X and +Z are defined relative to the lander, so they may be "tilted" with respect to the horizon)
- "**U**" - Untilt frame (special frame used for CYL-PER projection to compensate for Lander tilt on stereo panoramas)
- "**O**" - Other frame not listed above (if used often, additional letters will be added to the table for those frames)

brt = (1 alpha character) Specifies Brightness Correction flag indicating what kind of brightness (radiometric) seam-matching process has been applied to the mosaic. Brightness correction is a seam-matching process done on top of radiometric correction in order to make the mosaic look better; some radiometric accuracy is lost in the process. If multiple types of correction are done, the highest-level (last in the table) value is used.

NOTE: If any kind of brightness matching is applied, the inputs must first be radiometrically corrected. This is stated explicitly in the filename by <prod>, or implicitly if <prod> is not a radiometrically-corrected type, meaning that MIPLRAD has been used.

Valid values are:

- "_" - No brightness correction
- "**R**" - Radiometric correction only using MIPLRAD (no brightness seam matching)
- "**T**" - Radiometric correction only using TAMCAL (no brightness seam matching)
- "**B**" - Automatic brightness adjustment (multiplicative and/or additive factor applied to each frame)
- "**M**" - Manual brightness adjustment (multiplicative and/or additive factor applied to each frame)
- "**V**" - Anti-vignetting adjustment applied to some or all frames. Analogous to an additional flat field applied to the inputs to combat "ripple" effects caused by inadequate radiometric correction.
- "**O**" - Other method not listed above (if used often, additional letters will be added to this table accordingly)

act = (4 alphanumeric) Activity ID as extracted from round-trip accountability Token. Valid values are assigned at planning time by PSI tool.

pay = (1 alphanumeric) Identifier of Payload type as extracted from round-trip accountability Token. 1 Hex character representing 4 bits of the 32-bit Token based on a bit lookup, which follows:

Instrument	Bit String	Token Flag (<*observation type*>)	Valid Value
SSI, RAC	0000	Non-mosaic, Non-stereo (general science)	"**0**"
	0001	Non-mosaic, Non-stereo (calibration)	"**1**"
	0010	Non-mosaic, Non-stereo, Mesh/Reach (operations)	"**2**"
	0011	Non-mosaic, Non-stereo (atmospheric)	"**3**"
	0100	Non-mosaic, Stereo (general science)	"**4**"
	0101	Non-mosaic, Stereo (calibration)	"**5**"
	0110	Non-mosaic, Stereo, Mesh/Reach (operations)	"**6**"
	0111	Non-mosaic, Stereo (atmospheric)	"**7**"
	1000	Mosaic, Non-stereo (general science)	"**8**"

	1001	Mosaic, Non-stereo (calibration)	"**9**"
	1010	Mosaic, Non-stereo, Mesh/Reach (operations)	"**A**"
	1011	Mosaic, Non-stereo (atmospheric)	"**B**"
	1100	Mosaic, Stereo (general science)	"**C**"
	1101	Mosaic, Stereo (calibration)	"**D**"
	1110	Mosaic, Stereo, Mesh/Reach (operations)	"**E**"
	1111	Mosaic, Stereo (atmospheric)	"**F**"
MECA-OM	0000	Single, Single focus, monochrome (Clean)	"**0**"
	0001	Single, Single focus, monochrome (Exposed)	"**1**"
	0010	Single, Single focus, color (Clean)	"**2**"
	0011	Single, Single focus, color (Exposed)	"**3**"
	0100	Single, Through focus, monochrome (Clean)	"**4**"
	0101	Single, Through focus, monochrome (Exposed)	"**5**"
	0110	Single, Through focus, color (Clean)	"**6**"
	0111	Single, Through focus, color (Exposed)	"**7**"
	1000	Mosaic, Single focus, monochrome (Clean)	"**8**"
	1001	Mosaic, Single focus, monochrome (Exposed)	"**9**"
	1010	Mosaic, Single focus, color (Clean)	"**A**"
	1011	Mosaic, Single focus, color (Exposed)	"**B**"
	1100	Mosaic, Through focus, monochrome (Clean)	"**C**"
	1101	Mosaic, Through focus, monochrome (Exposed)	"**D**"
	1110	Mosaic, Through focus, color (Clean)	"**E**"
	1111	Mosaic, Through focus, color (Exposed)	"**F**"

spec = (1 alphanumeric) Special processing designator to be defined later. This character is used to indicate off-nominal or special processing of the image. Examples might be use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.

The meaning of any individual character in this field will be defined on an ad-hoc basis as needed during the mission. Within one Activity ID, the character will be used consistently, so this field can be used to group together all derived products resulting in one kind of special processing. An attempt will be made to maintain consistency across different Activity ID's as well, but this may not always be possible; thus the meaning of characters may change across different Activity ID's.

A text file will be maintained containing all special processing designators that are used, the Activity ID's they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.

Valid values are:

"_" - Nominal processing

"<*value*>" - Arbitrary value to be defined later that flags a specific type of Special processing

eye = (1 alpha character) Camera eye. Valid values are:

"**L**" - Left camera eye
"**R**" - Right camera eye
"**M**" - Monoscopic (non-stereo camera)
"**S**" - Stereo (2-banded image containing Left and Right eyes mapped to bands 1 and 2)
"**C**" - Color stereo (6-banded image containing color Left and Right images in the order: Lr, Lg, Lb, Rr, Rg, Rb)
"**A**" - 3-banded Anaglyph of Left, Right, Right eyes mapped to Red, Green, Blue channels
"**N**" - Not Applicable

filt = (3 alphanumeric) Specifies 3 spectral filter positions, LED settings, or color space components. For single-filter images, the filter code is repeated 3 times. For multiple-filter (color) images, the filters are listed in the order in which they appear in the file. A 2-band image would repeat the 2nd filter twice, while images with 4 or more bands list the first 2 filters with an X for the 3rd filter (exception: a 6-band color stereo image can list just 3 filters, if they are the same between left and right eyes).

Higher-order RDR processing can transform sets of raw filter/LED images into calibrated color images. These are represented in the standard xyY or sRGB color spaces. For the sRGB color space, the letters R,G,B are re-used, conflicting with LED states for RAC/OM or filter B for SSI. The difference should be obvious from context; if not, use the labels to distinguish.

Valid values:

Instrument	Filter Number	LED Status	Color Space
SSI	Hex	n/a	n/a
RAC	n/a	"**R**" = Red	n/a
	n/a	"**G**" = Green	n/a
	n/a	"**B**" = Blue	n/a
	n/a	"**D**" = Dark/off	n/a
	n/a	"**O**" = other (combination of multiple LED's)	n/a
MECA-OM	n/a	"**R**" = Red	n/a
	n/a	"**G**" = Green	n/a
	n/a	"**B**" = Blue	n/a
	n/a	"**D**" = Dark/off	n/a
	n/a	"**W**" = White	n/a
	n/a	"**U**" = UV	n/a
	n/a	"**O**" = other (combination of multiple LED's)	n/a
All	n/a	n/a	"**R**" = Red component of sRGB color space
	n/a	n/a	"**G**" = Green component of sRGB color space
	n/a	n/a	"**B**" = Blue component of sRGB color space
	n/a	n/a	"**X**" = x component of xyY color space
	n/a	n/a	"**Y**" = y component of xyY color space
	n/a	n/a	"**I**" = Y component of xyY color space (capital-I = Intensity)
	n/a	n/a	"**O**" = Other (if used often, additional letters will be added for other color spaces)

who = (1 alpha character) Product Creator indicator. Valid values are:

- "**A**" - University of **A**rizona
- "**M**" - **M**IPL (OPGS) at JPL
- "**C**" - **C**orrected (OPGS) manually at JPL
- "**O**" - MECA-**O**M Team
- "**T**" - **T**exas A&M (Mark Lemmon)
- "**X**" - Other

ver = (1 alphanumeric) Version identifier providing uniqueness for book keeping.

The valid values, in their progression, are as follows:

Range 1 thru 9 - **"1", "2"… "9"**
Range 10 thru 35 - **"A", "B" …"Z"**

The Version number increments by one whenever an otherwise-identical filename would be produced, independent of the Special field. Thus, even though the Special field may change, the Version number continues to increment by one over the previous Version. This allows the "best" Version of a product to be determined - irrespective of what special processing was done to achieve it - by simply looking for the highest Version number.

The following examples show how the Version field increments independently of the Special field thru a progression of RDR processing for the same SCLK product:

Examples		
Iteration of Processing	Special field value	Version field value
1	"_"	"1"
2	"B"	"2"
3	"B"	"3"
4	"E"	"4"
5	"_"	"5"
6	"A"	"6" (best)

ext = (2 to 3 alpha characters) Product type extension.

Valid values for nominal operations camera data products:

"**IMG**" - Camera image EDRs and RDRs (PDS labeled).
"**VIC** " - Camera image EDRs and RDRs with only VICAR label (no PDS label)

Valid values for quick-look JPEG compressed camera data products:

"**JPG**" - JPEG compressed (no PDS label).

Examples:

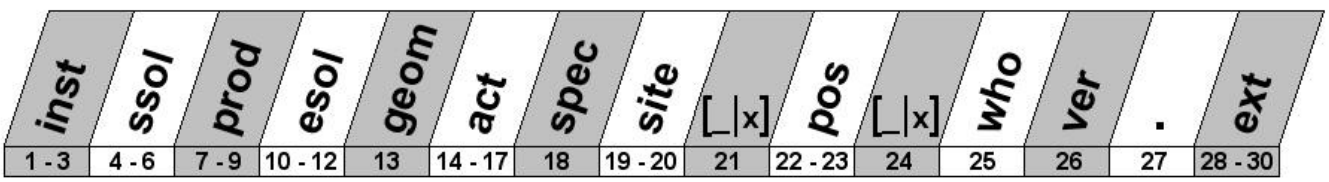
a) S_068RAL_CYP_P_002CC_R444M1.IMG — SSI instrument as primary ("S"), no instrument as secondary ("_"), Sol 68 ("068"), linearized Rad-corrected RDR as input ("RAL"), Mosaic filename field delimiter ("_"), Cylindrical-Perspective projection ("CYP"), no Geometry correction ("_"), Payload coordinate frame ("P"), no Brightness correction ("_"), Activity ID "002C" ("002C"), Mosaic/Stereo general science ("C"), Nominal processing ("_"), Right Eye ("R"), Filter position 4 ("444"), produced by MIPL ("M"), product version 1 ("1"), PDS-labeled ("IMG")

b) SR033FFL_CYLTLR003CFZLRGBM1.IMG — SSI instrument as primary ("S"), RAC instrument as secondary ("R"), Sol 33 ("033"), Full Frame Geom-corrected "linearized" RDR as input ("FFL"), Mosaic filename field delimiter ("_"), Cylindrical projection ("CYL"), manual Tiepointing form of geometry correction ("T"), Local Level coordinate frame ("L"), MIPLRAD mode of Brightness correction ("R"), Activity ID "003C" ("003C"), Mosaic/Stereo atmospheric ("F"), flag "Z" as an arbitrary type of Special processing determined by the product's Provider ("Z"), Left Eye ("L"), Red and Green and Blue components of sRGB color space as the 3 Filter positions ("RGB"), produced by MIPL ("M"), product version 1 ("1"), PDS-labeled ("IMG")

4.4.3 Terrain Mesh RDR

Each PHX Terrain Mesh RDR product can be uniquely identified by incorporating into the product filename the Lander Mission identifier, the Ending Sol identifier, the Instrument type(s) identifier, the Last Site location, the input Product Type identifier(s), the Lander's last Position within the last Site, and a Version number. The filename complies to the PDS 27.3 standard and is fixed length.

The filename convention follows:



where, "[ ]" denotes a required character selected from the following options: Underscore ("_") serving as a placeholder, or "**X**" denoting "multiple counts" of the previous field.

and where,

inst = (3 alpha characters) PHX science instrument identifier. If less than 3 Instruments are specified, then "_" characters are used to pad field to 3.

Valid values for Instrument identifiers:

"**S**" - SSI
"**R**" - RAC
"**H**" - HiRISE (MRO)
"**C**" - Context (general)

ssol = (3 integers) Specifies the starting Sol of acquired data contained in the Mesh. Zero padded to 3 digits. Example value is "004".

prod = (3 alpha characters) Product Type identifier of input data. Product types are differentiated as having camera-induced distortion removed ("linearized") or not removed (nominal):

Valid values for PHX camera instrument input data products:

Data Product	Non-linearized (NOMINAL)	Linearized
Full frame EDR	"**EFF**"	n/a
Sub-frame EDR	"**ESF**"	n/a
Downsampled EDR	"**EDN**"	n/a
Geometrically-corrected (linearized) Full frame RDR	n/a	"**FFL**"

Geometrically-corrected (linearized) Sub-frame RDR	n/a	"**SFL**"
Geometrically-corrected (linearized) Downsampled RDR	n/a	"**DNL**"
Slope RDR	"**SLP**"	"**SLL**"
Slope Heading RDR	"**SHP**"	"**SHL**"
Slope Magnitude RDR	"**SMP**"	"**SML**"
RA Reachability RDR	"**IDD**"	"**IDL**"

esol = (3 integers) Specifies the ending Sol of acquired data contained in the Mesh. Zero padded to 3 digits. Example value is "005".

geom = (1 alpha character) Specifies Geometry Correction flag. Indicates what kind of geometric correction has been applied to data. If multiple types of corrections were applied, then the highest-level (last in the table) value is used.

Valid values are:

- "_" - No correction (raw pointing)
- "**A**" - Auto-correction via tiepointing
- "**F**" - Auto-correction via tiepointing & auto-registration w/ fiducials
- "**G**" - Auto-correction via tiepointing & manual registration w/ fiducials
- "**T**" - Manual tiepointing
- "**R**" - Manual tiepointing & auto-registration w/ fiducials
- "**M**" - Manual tiepointing & manual registration w/ fiducials
- "**O**" - Other correction not listed above

act = (4 alphanumeric) Activity ID as extracted from round-trip accountability Token. Valid values are assigned at planning time by PSI tool.

spec = (1 alphanumeric) Special processing designator to be defined later. This character is used to indicate off-nominal or special processing of the image. Examples might be use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.

The meaning of any individual character in this field will be defined on an ad-hoc basis as needed during the mission. Within one Activity ID, the character will be used consistently, so this field can be used to group together all derived products resulting in one kind of special processing. An attempt will be made to maintain consistency across different Activity ID's as well, but this may not always be possible; thus the meaning of characters may change across different Activity ID's.

A text file will be maintained containing all special processing designators that are used, the Activity ID's they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.

Valid values are:

- "_" - Nominal processing
- "**X**" - Multiple Activities apply
- "<*value*>" - Arbitrary value to be defined later that flags a specific type of Special processing

site = (2 alphanumeric) Site location count. Use of both integers and alphas allows for a total range of 0 thru 1295. A value greater than 1295 is denoted by "##" (2 pound signs), requiring the user to extract actual value from label.

The valid values, in their progression, are as follows:

Range 0 thru 99 - **"00", "01", "02"… "99"**
Range 100 thru 1035 - **"A0", "A1" … "A9", "AA", "AB"…"AZ", "B0", "B1"… "ZZ"**
Range 1036 thru 1295 - **"0A", "0B"…"0Z", "1A", "1B"…"9Z"**
Range 1296 or greater - **"##"** (2 pound signs)

Example value is "AK" for value of 120.

pos = (2 alphanumeric) Position-within-Site count. Use of both integers and alphas allows for a total range of 0 thru 1295. A value greater than 1295 is denoted by "##" (2 pound signs), requiring the user to extract actual value from label.

The valid values, in their progression, are as follows:

Range 0 thru 99 - **"00", "01", "02"… "99"**
Range 100 thru 1035 - **"A0", "A1" … "A9", "AA", "AB"…"AZ", "B0", "B1"… "ZZ"**
Range 1036 thru 1295 - **"0A", "0B"…"0Z", "1A", "1B"…"9Z"**
Range 1296 or greater - **"##"** (2 pound signs)

Example value is "AK" for value of 120.

who = (1 alpha character) Product Creator indicator. Valid values are:

"**M**" - **M**IPL (OPGS) at JPL

ver = (1 alphanumeric) Version identifier providing uniqueness for book keeping.

The valid values, in their progression, are as follows:

Range 1 thru 9 - **"1", "2"… "9"**
Range 10 thru 35 - **"A", "B" …"Z"**

The Version number increments by one whenever an otherwise-identical filename would be produced, independent of the Special field. Thus, even though the Special field may change, the Version number continues to increment by one over the previous Version. This allows the "best" Version of a product to be determined - irrespective of what special processing was done to achieve it - by simply looking for the highest Version number.

The following examples show how the Version field increments independently of the Special field thru a progression of RDR processing for the same SCLK product:

Examples		
Iteration of Processing	Special field value	Version field value
1	"_"	"1"
2	"B"	"2"
3	"B"	"3"
4	"E"	"4"
5	"_"	"5"
6	"A"	"6" (best)

ext = (2 or 3 alpha characters) Product type extension.

Valid values are:

"**iv**" - Per-Wedge Terrain Mesh product in Inventor format (for MIPL use)
"**ht**" - Per-Wedge Height Map with VICAR label
"**pfb**" - Unified Terrain Mesh product in Performer Binary format
"**mod**" - Unified Height Map with VICAR label

Examples:

a) S__014EFF014_002C_01_01_M1.pfb — SSI instrument as primary ("S"), no instrument as secondary (“_”), no instrument as tertiary (“_”), Starting Sol 14 (“014”), Full Frame EDR as input (“EFF”), Ending Sol 14 (“014”), no geometric correction (“_”), Activity ID “002C” (“002C”), Nominal processing (“_”), Site 1 (“01”), Mesh filename field delimiter (“_”), Position 1 (“01”), Mesh filename field delimiter (“_”), produced by MIPL ("M"), product version 1 ("1"), Unified Terrain Mesh in Performer Binary format ("pfb")

b) SR_047FFL049T003B_01_03xM1.ht — SSI instrument as primary ("S"), RAC instrument as secondary (“_”), no instrument as tertiary (“_”), Starting Sol 47 (“047”), Full Frame Geom-corrected RDR as input (“FFL”), Ending Sol 49 (“049”), manual Tiepointing mode of geometric correction (“T”), Activity ID “003B” (“003B”), Nominal processing (“_”), Site 1 (“01”), Mesh filename field delimiter (“_”), Position 3 (“03”), Mesh filename field delimiter denoting multiple positions (“x”), produced by MIPL ("M"), product version 1 ("1"), Unified Height Map in VICAR format ("ht")

5. DETAILED DATA PRODUCT SPECIFICATIONS

5.1 EDR Data Products

The data packaged in the camera data files will be decoded, decompressed camera image data in single frame form as an Experiment Data Record (EDR). The Full Frame form of a standard image data file has the maximum dimensions of 1024 lines by 1024 samples. The other camera data files and their data sizes are listed in Table 5.1.

Table 5.1 - PHX Camera EDR Data Products

Image Type	Image Size	Pixel Size (bits)	Description
Full Frame	1024 lines x 1024 samples	16-bit signed integer	Nominal full sized, full resolution data product. Note that if “12 to 8-bit” scaling is commanded, then the valid pixels are stored as the last 8 bits of a 16-bit integer.
Sub-frame	variable	16-bit signed integer	Same format as Full Frame, but only a selected row (line) and/or column sub-frame is read back. The bit scaling rules described for the Full Frame case above also apply here.
Downsampled	variable	16-bit signed integer	In theory, images are converted to smaller images via a) nearest neighbor pixel averaging, or b) downsampling (sampling the upper right subsection).

Image Type	Image Size	Pixel Size (bits)	Description
			The bit scaling rules described for the Full Frame case above also apply here.
Reference pixels	1024 lines x 32 samples	16 (unsigned)	For SSI images only, these are Dark pixels bookending (pre- and post-) image pixels during serial register readout. There are 17 “pre-” Reference and 14 “post-” Reference pixels, plus 1 for the camera hardware serial number (left-shifted by 4 bits if 12-bit data).

5.1.1 Full Frame EDR

Full Frame EDRs are stored as 16-bit signed integers. If 12-to-8 bit scaling is performed, then pixels are stored in 16-bit format and only the last 8 bits of the 16-bit integer are used.

5.1.2 Sub-frame EDR

Sub-frame EDRs are a subset of rows and columns of the 1024x1024 full frame image. Sub-frame EDRs are stored as 16-bit signed integers. If 12-to-8 bit scaling is performed, then pixels are stored in 16-bit format and only the last 8 bits of the 16-bit integer are used.

5.1.3 Downsampled EDR

A downsampled EDR is a smaller version of the 1024x1024 full frame or subframed image using the following methods: a) nearest neighbor pixel averaging, or b) downsampling (sampling the upper right subsection). Downsampled EDRs are stored as 16-bit signed integers. If 12-to-8 bit scaling is performed, then pixels are stored in 16-bit format and only the last 8 bits of the 16-bit integer are used.

5.1.4 Reference Pixels

For the SSI camera, the onboard CCD array has 17 "pre-Reference" dark pixels (12-bits) located at the beginning and 14 "post-Reference" dark pixels (12-bits) located at the end of each row. Following the last "post-Reference" dark pixel, at the very end of each row, is the camera hardware serial number (left-shifted by 4 bits if 12-bit data).

5.2 RDR Data Products

The RDR data product is comprised of radiometrically decalibrated and/or camera model corrected and/or geometrically altered versions of the raw camera data, in both single and multi-frame (mosaic) form. Most RDR data products will have PDS labels, or if generated by MIPL (OPGS), dual PDS/VICAR labels. Non-labeled RDRs include JPEG compressed products and the Terrain products. The RDR data products that serve operational needs are listed in Table 5.2 below.

Table 5.2 - PHX Camera RDR Data Products

Data Product	# Bands	Data Type	Data Structure	PDS Sample Type
Radiometrically Corrected RDR a) SSI Rad-corrected RDR types: • RAD (SSI team) • IOF (SSI team) • RAD (OPGS - MIPLRAD) • RAD (OPGS - MIPLRAD2) • RAD (OPGS - MIPLRAD3) b) RAC Rad-corrected RDR types: • RAD (RAC team) • RAD (OPGS - MIPLRAD) • RAD (OPGS - MIPLRAD2) • RAD (OPGS - MIPLRAD3)	1	16-bit signed integer	PDS (science) or dual PDS/VICAR (OPGS) binary file.	MSB_INTEGER
CAHV Linearized RDR	1	16-bit signed integer	Dual PDS/VICAR (OPGS) binary file.	MSB_UNSIGNED_INTEGER or MSB_INTEGER
Disparity RDR	2	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Line Disparity RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Sample Disparity RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
XYZ RDR	3	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
X-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Y-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Z-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Range RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Surface Normal (UVW) RDR	3	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Surface Normal U-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Surface Normal V-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Surface Normal W-component RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
Surface Roughness RDR	1	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
RA Reachability RDR	12	16-bit signed	Dual PDS/VICAR	MSB_INTEGER

Data Product	# Bands	Data Type	Data Structure	PDS Sample Type
		integer or Float	(OPGS) binary file.	
Terrain Mesh RDR	3	Performer Binary (PFB)	PFB, no label	N/A
Mosaic RDR	1 or 3	16-bit signed integer or Float	PDS (science) or dual PDS/VICAR (OPGS) binary file.	MSB_INTEGER, IEEE_REAL or PC_REAL
Anaglyph RDR	3	Float	Dual PDS/VICAR (OPGS) binary file.	IEEE_REAL or PC_REAL
JPEG compressed RDR	1 or 3	8-bit unsigned	JPEG, no label	IEEE_REAL or PC_REAL

5.2.1 Radiometrically Corrected RDR

There are multiple methods of performing radiometric correction, distinguished by the RADIOMETRIC_CORRECTION_TYPE keyword. The most common are TAMCAL, RACCAL, MIPLRAD, MIPLRAD2, and MIPLRAD3.

5.2.1.1 TAMCAL Method (SSI Team)

This refers to radiometric correction of SSI instrument data only, performed by the SSI instrument team (Texas A&M University and University of Arizona) using their suite of software tools. It is the most precise correction method applicable to SSI data.

There are 2 general types of SSI Radiometrically-corrected RDR products that are generated by the SSI instrument team: Radiance-calibrated and Radiance-factor calibrated. Additional details on the radiometric processing and calibration of SSI images can be found in the SSI Calibration Report.

5.2.1.1.1 Radiance-calibrated RDRs ("RAD", “RAL”)

The non-linearized RDRs are generated from EDRs. They have all of the major instrumental/environmental calibrations applied, such as bias removal, dark current removal, electronic shutter smear effect removal, flat field correction, and bad pixel repair. Then they have been scaled to absolute radiance units using pre-flight radiometric calibration coefficients. The units on these files are (W/m^2/nm/sr).

An analogous RDR file type exists for the linearized (geometrically-corrected) SSI RDR as well, and it is labeled with the "RAL" product type identifier to correspond with the "RAD" type. In addition, floating point versions of this RDR may also be generated.

5.2.1.1.2 Radiance factor-calibrated RDRs ("IOF", “IOL”)

The non-linearized RDRs are generated from EDRs or "RAD" RDRs. They have all the major instrumental/environmental calibrations applied and have been scaled to absolute radiance units as described above, and then have been divided by the absolute radiance of the Sun at the top of the Martian atmosphere within the appropriate SSI bandpass, to generate radiance factor, or "I over F" values, where I is the radiance from the Martian scene and pi * F is the radiance from the Sun at the top of the Martian atmosphere (or on the surface, as determined by reflectance calibration targets. Since the solar radiance in the same units as the Mars scene radiance was divided out, these files are unitless but typically have values in the range of 0.0 to 1.0 (for example, average bright Mars soils exhibit I/F ~ 0.35 at 750 nm and I/F ~ 0.05 at 410 nm).

As with the "RAD" RDR type, there exists a linearized version of the IOF type of Radiometrically-corrected RDR, called "IOL". A floating point version of this RDR may also be generated.

5.2.1.2 RACCAL Method (RAC Team)

This refers to radiometric correction of RAC instrument data only, performed by the RAC instrument team (MPS) using their suite of RACCAL software tools. It is the most precise correction method applicable to RAC data. Note that radiometric correction of MECA-OM instrument data will be performed using the same tools employed for the RACCAL method.

The RAC/OM calibration steps performed by the RACSoft package, illustrated by Figure 5.2.1.2, are described below:

1. The bad pixel removal state replaces a number of pixels marked bad because of dust grains on the CCD or hot electron production. The bad pixels are replaced by an interpolated value based on the surrounding pixels.
2. The bias subtraction state subtracts the ADC digital offset from the image
3. The RAC and the OM uses an electronic shutter where the image data is fast clocked to a covered aread on the CCD at the end of the exposure. During the fast clocking each row experiences addition light from other parts of the scene. The electronic shutter correction subtracts from row N the summed DN signal of row 0 to N-1 scaled by the time it takes to clock a row one step on the CCD.
4. The dark current correction subtracts an estimated mean value of dark current based on the temperature of the CCD. This simple scheme (as compared to the SSI) is used because the RAC and OM has a very low dark current production under Mars conditions.
5. The flatfield correction divider the image by the relevant flatfield for the given focus motor step.
6. The OM calibration is finished after the flat field correction since good absolute calibration data is not available for the OM.
7. The final step of the RAC calibration is to divide the image by the absolute calibration constant for the given focus motor step. The calibration constant is given by the ground absolute calibration at focus motor step 306 (near infinite focus) and a correction factor derived for the change in instantaneous field of view between focus step 306 and the active focus step.

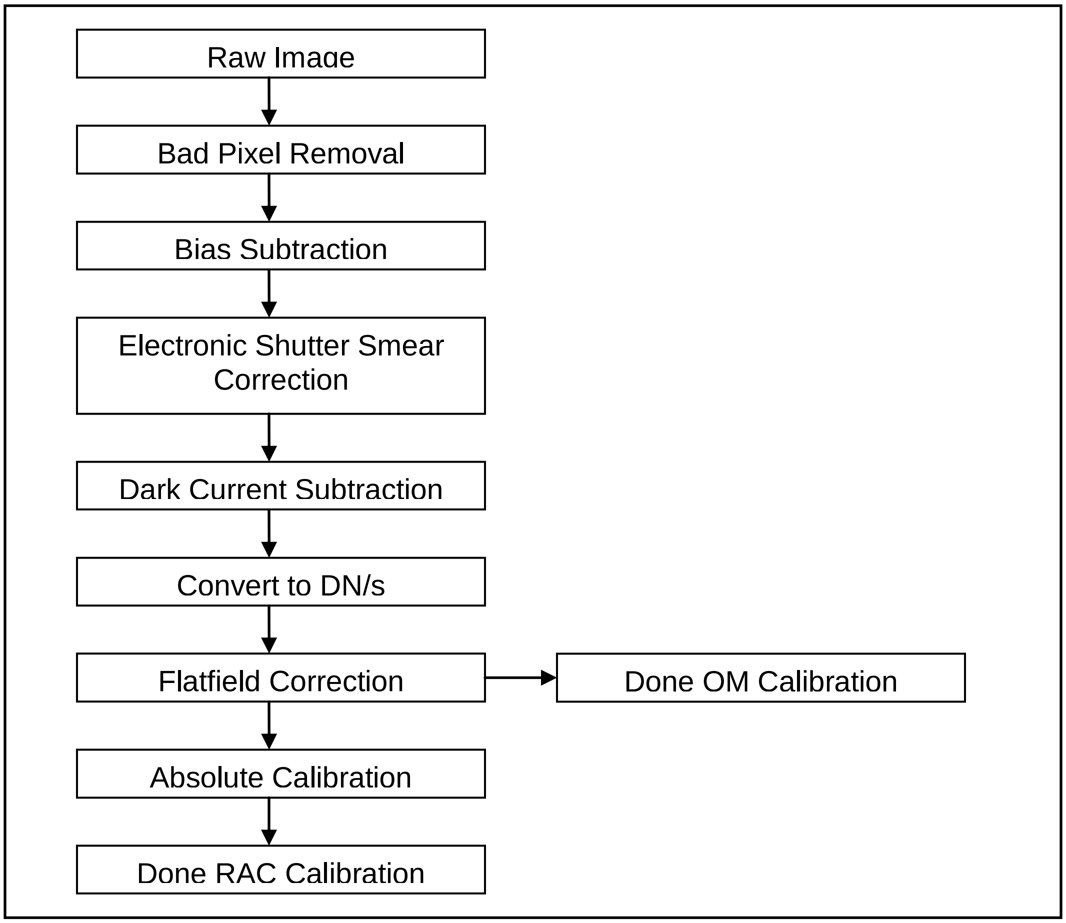


Figure 5.2.1.2 – RACSoft Data Flow

5.2.1.3 MIPLRAD, MIPLRAD2, MIPLRAD3 Methods (OPGS)

These refer to radiometric correction of any camera instrument data systematically performed by MIPL (OPGS at JPL) to meet tactical time constraints imposed by Arm Planners. The resulting rad-corrected RDRs are integrated into terrain mesh products used for RA trench digging. For SSI and RAC instrument data, these methods are less precise than the TAMCAL and RACCAL methods previously discussed. The **MIPL rad**iometrically-corrected RDR filenames carry the product type designators RAD (non-linearized) or RAL (linearized).

MIPLRAD, MIPLRAD2, and MIPLRAD3 are first-order corrections only and should be considered approximate. All three apply the following corrections: dark current, temperature-compensated responsivity, exposure time, binning correction, and flat field. The result is calibrated to physical units for PHX of W/m^2/nm/sr. The actual algorithm and equations used for the three MIPLRAD's are shown below. In all cases, ALL_CAPITALS serve to denote keyword names in the PDS label.

The only difference between the three MIPLRAD methods is in the dark current calculation that is used. MIPLRAD uses a dark current calculation developed by Adam Shaw at the University of Arizona.

MIPLRAD2 (the default) uses a calculation developed by Mark Lemmon at Texas A&M University. MIPLRAD3 uses the Lemmon calculation with a simplification for efficiency (described below).

Dark current applies only to SSI. RAC dark current is assumed to be 0 in all three methods.

1. Shaw Model Dark Current (MIPLRAD):

$$T_p = \text{INSTRUMENT\_TEMPERATURE(6)} \quad \{\text{SSI\_PCB\_END}\}$$
$$T_e = \text{INSTRUMENT\_TEMPERATURE(1)} \quad \{\text{SSI\_CCD\_END}\}$$
$$\text{if unavailable, } T_e = T_p - 20.0$$
$$T_s = \text{INSTRUMENT\_TEMPERATURE(0)} \quad \{\text{SSI\_PCB\_START}\}$$
$$\text{if unavailable, } T_s = T_p - 20.0$$
$$T_{ave} = \frac{T_s + T_e}{2}$$
$$\Delta\text{Bias} = \frac{1}{2}(4095 - \text{OFFSET\_NUMBER})$$
$$\text{mean\_bias} = b_0 + b_1 \, \mathrm{e}^{-b_2 / (T_p + 273.15)} + \Delta\text{Bias}$$
$$\text{storage\_area\_dc\_factor} = c_0 \, \mathrm{e}^{-c_1 / (T_e + 273.15)}$$
$$\text{active\_area\_dc\_factor} = \frac{\text{EXPOSURE\_DURATION}}{1000} \beta_0 \, \mathrm{e}^{-\beta_1 / (T_{ave} + 273.15)}$$

for each row i

$$r = 1024 - i$$
$$\text{bias}_r = \text{mean\_bias} + a_0 + a_1 (r + \mathit{offset})^{a_2}$$

for each column j

$$T_{\min} = \max\left(T_e, \mathbf{LowPoints}_{i,j}\right)$$
$$\text{storage\_dc}_{i,j} = \mathbf{Q_0}_{i,j} + \mathbf{Q_1}_{i,j} T_{\min} + \mathbf{Q_2}_{i,j} T_{\min}^2$$
$$\text{active\_dc}_{i,j} = \mathbf{H}_{i,j} \text{active\_area\_dc\_factor}$$
$$\text{dark}_{i,j} = \text{bias}_r + \text{storage\_dc}_{i,j} + \text{active\_dc}_{i,j}$$

where $a_n, b_n, c_n, \beta_n, \mathit{offset}$ are scalar calibration parameters and $\mathbf{H}, \mathbf{Q_n}, \mathbf{LowPoints}$ are image calibration parameters. Note that the row and column numbers are adjusted to compensate for subframing and downsampling, since the calibration files and row number are based on a full-size image. For downsampled images, the calibration files are averaged over the downsample area (similar to flat fields, below).

2. Lemmon Model Dark Current (MIPLRAD2 and MIPLRAD3):

$$T_p = \text{INSTRUMENT\_TEMPERATURE(6)} \quad \{\text{SSI\_PCB\_END}\}$$
$$T_e = \text{INSTRUMENT\_TEMPERATURE(1)} \quad \{\text{SSI\_CCD\_END}\}$$
$$\text{if unavailable, } T_e = T_p - 20.0$$
$$T_s = \text{INSTRUMENT\_TEMPERATURE(0)} \quad \{\text{SSI\_PCB\_START}\}$$
$$\text{if unavailable, } T_s = T_p - 20.0$$

$$t_c = 70/0.00512 \quad \{70 \text{ sec. in counts}\}$$

$$t_{\exp} = \text{EXPOSURE\_DURATION}/0.00512$$

$$\Delta T_0 = \frac{T_e - T_s}{\left(e^{\frac{-(t_{\exp}+1000.)}{t_c}} \right) - 1.0} \quad \{+1000 \text{ compensates for storage readout}\}$$

$$T_{\inf} = T_s - \Delta T_0$$

$$\overline{T} = \frac{\sum_{i=0}^{20} T \times w}{\sum_{i=0}^{20} w}$$

$$\text{where } T = T_{\inf} + \Delta T_0 e^{-\frac{i \times t_{\exp}}{20 t_c}}$$

$$\text{and } w = \left(\frac{T + 273.15}{273.15} \right)^{1.5} e^{-7310\left(\frac{1}{T+273.15} - \frac{1}{273.15} \right)}$$

$$\Delta\text{Bias} = \frac{1}{2}\left(4095 - \text{OFFSET\_NUMBER}\right)$$

for each row *i*

$$\text{bias}_i = \Delta\text{Bias} + b_0 + b_1 \, \mathrm{e}^{-b_2\left(\frac{1}{T_p + 273.15} - \frac{1}{273.15} \right)} + \mathbf{bias\_col}_i$$

for each column *j*

$$\text{storage\_dc}_{\text{i,j}} = \mathrm{B}_{\mathbf{0}_{\mathbf{i,j}}} e^{-\beta_1\left(\frac{1}{T_e + 273.15} - \frac{1}{273.15} \right)}$$

$$X = \frac{1}{\overline{T}} - \frac{1}{273.15}$$

MIPLRAD2 method :

$$\text{active\_dc}_{i,j} = \Gamma_{\mathbf{0}_{i,j}} \left(\overline{T}/273.15 \right)^{1.5} e^{\Gamma_{\mathbf{1}_{i,j}} X + \Gamma_{2_{i,j}} X^2} \times \text{EXPOSURE\_DURATION}$$

MIPLRAD3 method :

$$\text{active\_dc}_{i,j} = \Gamma_{\mathbf{0}_{i,j}} \left(\overline{T}/273.15 \right)^{1.5} e^{\gamma_1 X + \gamma_2 X^2} \times \text{EXPOSURE\_DURATION}$$

$$\text{dark}_{i,j} = \text{bias}_i + \text{storage\_dc}_{i,j} + \text{active\_dc}_{i,j}$$

where b_n, β_1, γ_n are scalar calibration parameters, $\mathbf{bias\_col}$ is a 1x1024 parameter array, and $\mathrm{B}_{\mathbf{0}}, \Gamma_{\mathbf{0}}, \Gamma_{\mathbf{1}}, \Gamma_{\mathbf{2}}$ are image calibration parameters. Note that the row and column numbers are adjusted to compensate for subframing and downsampling, since the calibration files and row number are based on a full-size image. For downsampled images, the calibration files are averaged over the downsample area (similar to flat fields, below).

3. Temperature responsivity:

for SSI:

$$\text{resp} = \frac{1}{r_0 + r_1 \overline{T} + r_2 \overline{T}^2}$$

where $\overline{T}$ is as defined above for Lemmon dark current except that $w = 1.0$, and r_n are scalar calibration parameters.

for RAC:

$$T_{opt} = \text{INSTRUMENT\_TEMPERATURE(1)} \quad \{\text{RAC\_OPTICAL\_BENCH}\}$$

$$T = \left(T_{opt} - temp\_term\right) / temp\_factor \quad \{\text{convert to counts}\}$$

$$R = \left(r_0 + r_1 T + r_2 T^2\right) 1000 \quad \{\text{convert from } \mu\text{m to nm}\}$$

$$f = \text{ARTICULATION\_DEVICE\_COUNT(0)} \quad \{\text{FOCUS\_POS - CURRENT}\}$$

$$ifov = \tan^{-1}\left(\frac{focus\_scale}{focus\_Xi - (312 - f)\, focus\_delta}\right)$$

$$ifov_{ref} = \tan^{-1}\left(\frac{focus\_scale}{focus\_Xi - (312 - 6)\, focus\_delta}\right)$$

$$resp = R\left(\frac{ifov}{ifov_{ref}}\right)^2$$

4. Exposure time::

$$\exp = \frac{\text{EXPOSURE\_DURATION}}{1000}$$

5. Flat field:

$$n = \text{PIXEL\_AVERAGING\_WIDTH}$$

$$m = \text{PIXEL\_AVERAGING\_HEIGHT}$$

$$\text{flat}_{\text{i,j}} = \frac{1}{nm} \sum_{x=0}^{n} \sum_{y=0}^{m} \mathbf{FLAT}_{i+x,\, j+y}$$

i.e., the average flat field over a downsampled area. The image locations are adjusted for subframing as well, to pick up the appropriate place in the **FLAT** file.

6. Binning compensation:

This partially compensates for a flight software bug in computing downsampled images. It is an approximation which works for most of the image, but the first few columns are still wrong. It uses a factor which is based on the binning mode as follows:

Mode	Full-scale	2x2	3x3	4x1	1x4	4x4SW	4x4HW
Factor	1.0	4.0	2.25	1.3333333	4.0	1.7777778	5.3333333

7. Final correction

The final correction is then computed:

$$\text{output\_rad}_{i,j} = \frac{\left(\text{input\_image}_{i,j} - \text{dark}_{i,j}\right)}{\text{flat}_{i,j} \times \exp \times \text{binning\_correction} \times \text{resp}}$$

$$\text{output\_image}_{i,j} = \text{int}\left(\frac{\text{output\_rad}_{i,j} - \text{RADIANCE\_OFFSET}}{\text{RADIANCE\_SCALING\_FACTOR}} + 0.5\right)$$

output_rad is in radiance units, while RADIANCE_OFFSET and RADIANCE_SCALING_FACTOR are used to convert those into integers in output_image for ease of manipulation.

5.2.2 Geometrically Corrected RDR

EDRs and single-frame RDRs are described by a camera model. This model, represented by a set of vectors and numbers, permit a point in space to be traced into the image plane, and vice-versa. There are multiple camera models applicable in performing geometric correction of image data, and they largely are implemented by two parties: MIPL (for SSI and RAC instruments) and the MECA-OM team (for MECA-OM instrument).

5.2.2.1 MECA-OM Geometric Correction

MECA optics have been optimized for a flat surface located at the optimal, fixed image plane, and no geometric correction is provided. A minor distortion may result from substrate tilt across the 1-mm wide field of view as the substrate is rotated around the center of the 57.3 mm radius sample wheel. The MECA OM is designed to evaluate particle shape and size for particles smaller than 0.1 mm, on which scale this distortion is negligible. A small vertical movement of the image accompanies Sample Wheel and Translation Stage (SWTS) travel in and out of focus, and must be compensated when preparing through-focus image products.

5.2.2.2 MIPL Geometric Correction

EDR camera models are derived by acquiring images of a calibration target with known geometry at a fixed azimuth/elevation. The vectors representing the model are derived from analysis of this imagery. These vectors are then translated and rotated based on the actual pointing of the camera to represent the conditions of each specific image. An additional (small) pointing correction is applied based on temperature. This is a quadratic function which affects the rotation and scaling of the camera models. The results are the "camera model" for the EDR.

The SSI and RAC use a CAHVOR model, which is nonlinear and involves some complex calculations to transform line/sample points in the image plane to XYZ positions in the scene. To simplify this, the images are "warped", or reprojected, such that they can be described by a linear CAHV model. This linearization process has several benefits:

1) It removes geometric distortions inherent in the camera instruments, with the result that straight lines in the scene are straight in the image.
2) It aligns the images for stereo viewing. Matching points are on the same image line in both left and right images, and both left and right models point in the same direction.
3) It facilitates correlation, allowing the use of 1-D correlators.
4) It simplifies the math involved in using the camera model.

However, it also introduces some artifacts in terms of scale change and/or omitted data (see the references).

The linearized CAHV camera model is derived from the EDR's camera model by considering both the left and right eye models and constructing a pair of matched linear CAHV models that conform to the above criteria. For details on this algorithm see the references.

The image is then projected, or warped, from the CAHVOR/CAHVORE model to the CAHV model. This involves projecting each pixel through the EDR camera model into space, intersecting it with a surface, and projecting the pixel back through the CAHV model into the output image.

- C - The 3D position of the entrance pupil
- A - A unit vector normal to the image plane pointing outward (towards C)
- H - A vector pointing roughly rightward in the image; it is a composite of the orientation of the CCD rows, the horizontal scale, the horizontal center
- V - A vector pointing roughtly downward in the image; it is a composite of the orientation of the CCD columns, the vertical scale, the vertical center, and A.

If P is a point in the scene then the corresponding image locations x and y can be computed from:

$$x = \frac{(P - C)H}{(P - C)A}$$

$$y = \frac{(P - C)V}{(P - C)A}$$

For details on the camera model math and calibration and more description of the CAHV-model family, see references [Ref 8] through [Ref 17].

5.2.3 Disparity RDR

A Disparity file contains 2 bands of 32-bit floating point numbers in the Band Sequential order (line, sample). Alternatively, line and sample may be stored in separate single-band files.

The parallax, or difference measured in pixels, between an object location in two individual images (typically the left and right images of a stereo pair) is also called the “disparity”. Disparity files contain these disparity values in both the line and sample dimension for each pixel in the reference image. This reference image is traditionally the left image of a stereo pair, but could be the right image for special products. The geometry of the Disparity image is the same as the geometry of the reference image. This means that for any pixel in the reference image the disparity of the viewed point can be obtained from the same pixel location in the Disparity image.

The values in a Disparity image are the 1-based coordinates of the corresponding point in the non-reference image. Thus, the coordinates in the reference image are the same as the *coordinates* in the Disparity image, and the matching coordinates in the stereo partner image are the *values* is the Disparity image. Disparity values of 0.0 indicate no valid disparity exists, for example due to lack of overlap or correlation failure. This value is reflected in the MISSING_CONSTANT keyword.

5.2.3.1 Stereo Pair Matching Method
Inherent in the designed operation of the SSI instrument are time offsets between the acquisition of left and right images intended for stereo processing. So, the SCLK timestamp will not be reliable to automatically identify a stereo image pair, as opposed to the SCLK-matching strategy used to identify stereo image pairs for the Mars Exploration Rover (MER) mission.

For the PHX mission, MIPL will rely on two criteria available in the data product metadata label for automatic matching of left and right images comprising a stereo pair. First, any image product that was commanded as part of a stereo pair will have a value of "STEREO" for the label keyword "FRAME_TYPE". Second, the Activity ID component of the data product's 32-bit "token" value for each image in a stereo pair will be identical. The Activity ID value is preserved in the label keyword "OPS_TOKEN_ACTIVITY" (see Appendices A and B), as well as in a field embedded in the data product's filename nomenclature (see Section 4.4).

5.2.4 XYZ RDR
An XYZ file contains 3 bands of 32-bit floating point numbers in the Band Sequential order. Alternatively, X, Y and Z may be stored in separate single-band files as a X Component RDR, Y Component RDR and Z Component RDR, respectively. The single component RDRs are implicitly the same as the XYZ file, which is described below. XYZ locations in all coordinate frames for PHX are expressed in meters unless otherwise noted.

The pixels in an XYZ image are coordinates in 3-D space of the corresponding pixel in the reference image. This reference image is traditionally the left image of a stereo pair, but could be the right image for special products. The geometry of the XYZ image is the same as the geometry of the reference image. This means that for any pixel in the reference image the 3-D position of the viewed point can be obtained from the same pixel location in the XYZ image. The 3-D points can be referenced to any of the PHX coordinate systems (specified by DERIVED_IMAGE_PARAMS Group in the PDS label).

Most XYZ images will contain "holes", or pixels for which no XYZ value exists. These are caused by many factors such as differences in overlap and correlation failures. Holes are indicated by X, Y, and Z all having the same specific value. This value is defined by the MISSING_CONSTANT keyword in the IMAGE object. For the XYZ RDR, this value is (0.0,0.0,0.0), meaning that all three bands must be zero (if only one or two bands are zero, that does not indicate missing data).

5.2.4.1 XYZ Lander Volume Exclusion Mask
For the purposes of Terrain Mesh RDR generation, OPGS will create "Lander Volume Exclusion Mask" files that can be applied to the XYZ RDR. The Mask files are internal OPGS files not intended for archive or use by others, but they are stored in the OSS in the same directories as the XYZ RDRs. They are used to filter out lander and/or RA features from generated terrain products. They are single-band, byte files corresponding to (and derived from) an XYZ image, where 255 indicates the corresponding pixel is on the Lander and should be removed, or 0 indicates the pixel should remain in the output. The contents of the label are explicitly not defined, but in practice it is usually a copy of the XYZ image's label. These Mask files will assume the 3-character Product Type values "MSK" (Full Frame, Sub-frame, Downsampled) and "MSL" (linearized "MSK").

5.2.4.2 Masked XYZ Image
The XYZ (or XYL) image will be combined with the volume exclusion mask to create a Masked XYZ image. The contents are identical to the XYZ (XYL) image where the mask is a 0, but contain (0,0,0) (indicating no data) where the mask is 255. The 3-character Product Type values are "XMZ" (non-linearized) and "XML" (linearized).

5.2.5 Range RDR

A Range (distance) file contains 1 band of 32-bit floating point numbers.

The pixels in a Range image represent Cartesian distances from a reference point (defined by the RANGE_ORIGIN_VECTOR keyword in the PDS label) to the XYZ position of each pixel (see XYZ RDR). This reference point is normally the camera position as defined by the C point of the camera model. A Range image is derived from an XYZ image and shares the same pixel geometry and XYZ coordinate system. As with XYZ images, range images can contain holes, defined by MISSING_CONSTANT. For PHX, this value is 0.0.

5.2.6 Surface Normal (UVW) RDR

A Surface Normal (UVW) file contains 3 bands of 32-bit floating point numbers in the Band Sequential order. Alternatively, U, V and W may be stored in separate single-band files as a U Component RDR, V Component RDR and W Component RDR, respectively. The single component RDRs are implicitly the same as the UVW file, which is described below.

The pixels in a UVW image correspond to the pixels in an XYZ file, with the same image geometry. However, the pixels are interpreted as a unit vector representing the normal to the surface at the point represented by the pixel. U contains the X component of the vector, V the Y component, and W the Z component. The vector is defined to point out of the surface (e.g. upwards for a flat ground). The unit vector can be referenced to any of the PHX coordinate systems (specified by the DERIVED_IMAGE_PARAMS Group in the PDS label).

Most UVW images will contain "holes", or pixels for which no UVW value exists. These are caused by many factors such as differences in overlap, correlation failures, and insufficient neighbors to compute a surface normal. Holes are indicated by U, V, and W all having the same specific value. Unlike XYZ, (0,0,0) is an invalid value for a UVW file, since they're defined to be unit vectors. Thus there's no issue with the MISSING_CONSTANT as there is with XYZ, where (0.0,0.0,0.0) is valid.

5.2.7 Surface Roughness RDR

The roughness map contains surface roughness estimates at each pixel in an XYZ image. The roughness is computed as the maximum peak-to-peak deviation from the local plane. Units are meters; that is, a pixel value of 0.05 means that the local surface about that pixel has a maximum peak-to-peak deviation along the surface normal by 0.05m (5cm). Roughness values above some useful threshold (maximum roughness) are clipped to that threshold. If a roughness could not be computed for a pixel (e.g. because of lack of range data, or too much noise in the range data), then the roughness value at that pixel will be set to the "bad roughness" value (which must be greater than maximum roughness).

5.2.8 RA Reachability RDR

An RA Reachability map contains information about whether or not the instruments on the RA can "reach" (contact or image) the object or location represented by each pixel in the scene. It is derived from the XYZ and Surface Normal (UVW) products.

The geometry of the reachability map matches the linearized reference, XYZ, and Surface Normal (UVW) images, in that each pixel in the file directly corresponds to the pixel at the same location in the other products.

The reachability map is a 12-band byte image in standard Band Sequential order. Thus for each pixel there are 12 values. These values represent reachability for each of the 6 RA instrument placements in each of its 2 configurations. The mapping between band number and instrument/configuration is given by the INSTRUMENT_BAND_ID and CONFIGURATION_BAND_ID labels, and is summarized in Table 5.2.6.

Table 5.2.6 - RA Reachability Band Assignments

RA Tool	RA Configuration	
	ELBOW_UP	ELBOW_DOWN
SCOOP	Band 1	Band 2
SCOOP_BTM	Band 3	Band 4
BLADE	Band 5	Band 6
ISAD1	Band 7	Band 8
ISAD2	Band 9	Band 10
TECP	Band 11	Band 12

The value of the pixel is interpreted according to the instrument. For SCOOP, 0 means the pixel is not reachable in that configuration, while any other number represents the maximum preload in integer Newtons that can be applied at that point. For all other instruments, 0 means the pixel is not reachable by that instrument in that configuration, while 255 means that the pixel is reachable.

5.2.9 Terrain Map RDR

Terrain models are a high level product which are derived from the XYZ files and the corresponding image files. The terrain models are generated by meshing or triangulating the XYZ data based on the connectivity implied by the pixel ordering or by a volume based surface extraction. The XYZ files can be viewed as a collection of point data while the terrain models take this point data and connect it into a polygonal surface representation. The original image is referenced by the terrain models as a texture map which is used to modulate the surface color of the mesh. In this way the terrain models can be viewed as a surface reconstruction of the ground near the instrument with the mesh data capturing the shape of the surface and the original image, applied as a texture map, capturing the brightness variations of the surface. Specific terrain model formats such as VST, PFB, DEM and others can be viewed as analogous to GIF, TIFF or VICAR in image space in that each represents the data somewhat differently for slightly different purposes.

IMPORTANT NOTE: Terrain Map RDRs generated in support of Phoenix mission operations will not be archived to PDS, because they lack the necessary PDS metadata labels.

5.2.9.1 PFB Terrain Mesh

The Performer Binary (PFB) format facilitates the representation of a terrain surface as polygons, optimized for use by the RSVP tool. The number of polygons at any one time may vary according to site specific features, such as small rocks versus large boulders.

5.2.9.2 Terrain Mesh Height Map

For each unified Terrain Mesh there is an associated Height Map in VICAR format that serves as a digital elevation map.

5.2.10 Mosaic RDR

This section discusses the process of mosaicking multiple frames into a single RDR product. The text largely reflects the methods applied by MIPL under OPGS, associating projections with the mosaicking process. It should be noted that these processes can be independent, and that governing methods and software can differ between OPGS and the other science instrument teams. For instance, it is possible that OPGS and other science teams' software will transform individual images to one of the projections discussed below, without involving any mosaicking. Detailed mathematical descriptions of the mosaic projections and algorithms will be available in a separate paper "Mars Mosaic Projection Algorithms".

5.2.10.1 Overview of Mosaics in General

Mosaics can be assembled autonomously by tracing a view ray from each mosaic location or pixel into the scene, determining its intersection with a ground plane, and then querying each input image to determine if that point lies within its field of view. In this fashion mosaics containing several hundred images can be assembled for each spectral band in about 3 minutes each. It may be necessary to refine the camera pointing in order to produce accurate mosaics. This requires the determination of the actual azimuth and elevation of each image in order to correct for errors such as gear backlash. One way to do this is to acquire tiepoints between all pairs of overlapping images. Camera pointing commands are then estimated which cause the camera model to map the tiepoints to their correct locations. In some cases this can be accomplished automatically, but in general it requires human intervention to select tiepoints because of nonexistent overlap or changing lighting.

There are many uses for the image mosaics. First is the assembly of small pieces into a larger field of view. This includes tilting the camera model in the Mars coordinate system to model a tilted spacecraft, which should result can in mosaics with a level horizon beginning and ending at Mars north. The science teams can use these products to orient themselves. Another application is to provide to the RA planning team each Sol a small stereo mosaic which is registered to a fixed reference image. This permits the triangulation of way points for the next Sol's maneuvering of the RA instrument for trench digging.

For mosaicking of MECA-OM data, the only degree of mechanical freedom is rotation of the sample in the focal plane. Mosaics consist of adjacent areas on sample substrates stitched together by adjusting the overlap. Thus all mosaics will be 2-mm tall (512 rows) and an arbitrary width unless the component images are subframed in the vertical direction. Extremely minor stitching errors may result from left-right tilt associated with rotation (see Geometric Corrections), but this is not considered significant for the purposes of particle analysis.

5.2.10.2 How MIPL Creates Mosaics

The process used by the MIPL software to create mosaics is described below. It consists of several sub-steps. Conceptually, one can think of the process as adjusting the inputs, projecting them down to a surface, and looking at the result from a different point of view (the output projection). In reality, the process is run in reverse for ease of interpolation (this is described below).

A. *Input Pointing Adjustment* - There are several methods by which improved pointing of the cameras can be determined. The most common method is to pick tiepoints between image pairs and use that in a global function minimization to determine the corrected pointing parameters. Pointing parameters can also be determined manually.

   Regardless of method, the result is encapsulated in a "nav" file. A complete description of this file is outside the scope of this SIS, but fundamentally, this file contains, for each image being corrected, the original pointing parameters, and the improved pointing parameters.

   Pointing parameters are simply those numbers which represent how the camera is pointing in the Payload frame, reduced to the available degrees of freedom. These are used as inputs to the kinematics procedures which derive the camera model. Thus for the SSI attached to the mast, there are two parameters: azimuth and elevation actuator angles. In a departure from previous practice (e.g. MER), the RAC does not use joint angles but instead dispenses with kinematics calculations entirely, using explicit position and orientation of the camera instead (note that orientation can be specified either via a quaternion or as a set of Euler angles).

   The MIPL procedure exactly duplicates the on-board flight software mechanism for doing this (even using some flight code). Therefore the process is not described in more detail here. Suffice it to say that, in the absence of any change to the pointing, the result is the exact same camera model that the flight software generates and that ends up in the image label.

   The MIPL code allows for an extra pointing parameter for the SSI, called "twist", to characterize an additional degree of freedom. This has the effect of taking the final camera model, and rotating it by the specified amount around the A axis (A being one of the CAHV camera model parameters).

   Note that this entire process is completely optional; the telemetry camera model can be used as-is.

   Regardless of how obtained, the final result is a CAHV, CAHVOR, or CAHVORE camera model that describes the input geometry.

B. *Output Projection Determination* - The output projection is then determined. The parameters describing the projection are listed in Appendix A, and described in detail in Appendix B. The output projection parameters are determined by analysis of the inputs to give the "best" resulting mosaic, but can be overridden by the user. The determination process is outside the scope of this document; the results are what is important and they are in the label.

C. *Surface Determination* - A surface model is critical for mosaics. This is a mathematical surface which hopefully matches the actual scene. To the extent that the scene differs from the surface model, distortion and uncorrectable seams can result.

   Usually the surface model is a flat plane, with normal pointing upwards. This can be adjusted, however, to better match the scene. Regardless, the results are documented in the SURFACE_MODEL_PARMS group.

   There are five potential surface models in the MIPL software: PLANE, SPHERE2, INFINITY, SPHERE1, and SPHERE. To date, only the first two have been used. See SURFACE_MODEL_TYPE in Appendix B for description.

D. *Computation of Output View Ray* - For each pixel in the output mosaic, a view ray in 3-D space is constructed. How this view ray is constructed depends on the projection type. Below, the pixel is at location (i,j) in 0-based coordinates, with i corresponding to sample and j to line. (0,0) is in the upper-left-hand corner. Capitalized values represent PDS label items from the SURFACE_PROJECTION_PARMS group. Unit and coordinate system conversions are applied as necessary but are not specified here. The coordinate system used is defined by REFERENCE_COORD_SYSTEM_* in SURFACE_PROJECTION_PARMS.

E. *Projection from Output to Surface* - Once the view ray is determined, it is projected out until it intersects with the surface model. The resulting point in XYZ space is used in the next step. If the ray does not intersect the surface, the point is assumed to be at infinity in the direction the view ray is pointing. Exception: as mentioned above, the Vertical projection will reverse the direction of its view ray; infinity is assumed only if they both miss.

   Note that the INFINITY surface model guarantees the ray will miss the surface at all times.

   The difference between the SPHERE1 and SPHERE2 models is that, if the ray intersects the spherical surface more than once, SPHERE1 will take the first intersection, while SPHERE2 will take the second. For normal Lander situations, SPHERE1 thus roughly models a hill, while SPHERE2 roughly models a crater.

F. *Projection from Surface to Input* - The XYZ location (or direction for the infinity case) is then back-projected into each input camera model in turn, using the corresponding input camera model. The first input for which the resulting pixel coordinate is inside the image (excluding border pixels which are thrown away) stops the process; that is the image from which the output pixel value is taken.

   Note that this has the effect of stacking the images such that the first one in the input list of images "wins". There is no feathering of overlaps; the first image is "on top" of all the others, and an image completely covered by preceding images will not be used at all.

G. *Interpolation and Storage of the Result* - Finally, a bilinear interpolation is performed on the input image, based on the 4 pixels surrounding the back-projected location. The result of this interpolation is the value of the output pixel.

   Bilinear interpolation is optional, and is normally not done when making mosaics of XYZ or Surface Normal (UVW) data.

5.2.10.3 Cylindrical Projection Mosaic

The MIPL method for creating a Cylindrical projection involves computing the azimuth and elevation of the view ray, as follows:

azimuth = i / MAP_RESOLUTION + START_AZIMUTH

elevation = (ZERO_ELEVATION_LINE - j) / MAP_RESOLUTION

The view ray emanates from the point PROJECTION_ORIGIN_VECTOR.

Figure 5.2.11.1 shows such a mosaic overlaid onto azimuth and elevation grid lines, with individual frame boundaries superimposed and annotated by number. In this case each pixel represents a fixed angle in azimuth and elevation. Rows are of constant elevation in Mars coordinates. The horizon is level, and columns begin clockwise from Mars north.

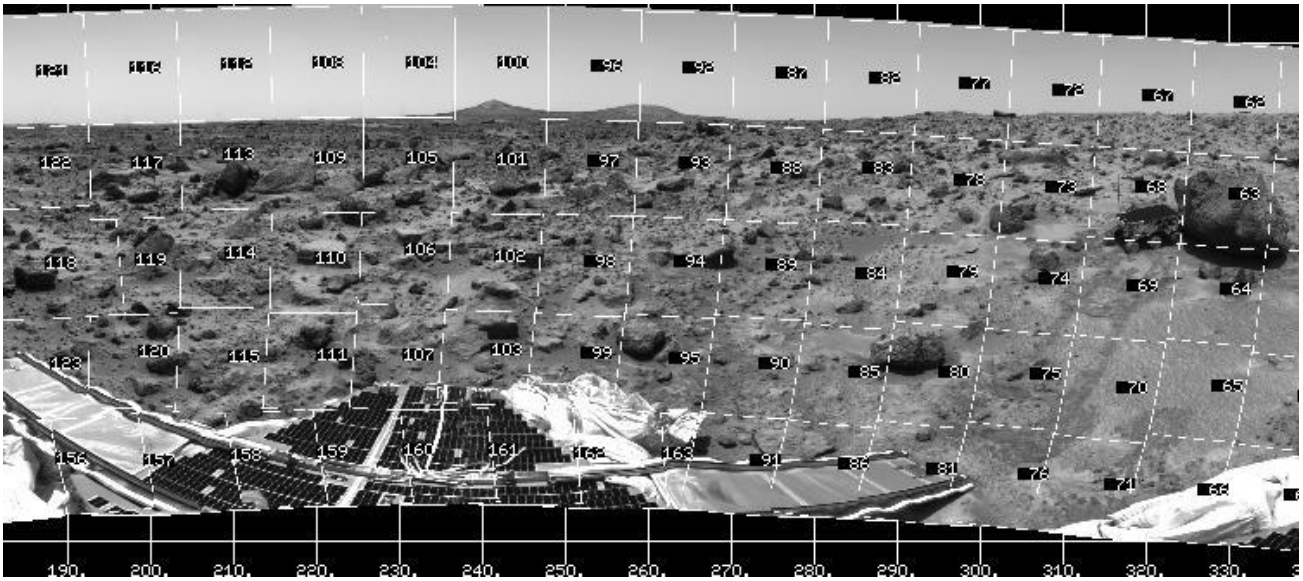


Figure 5.2.11.1 – Cylindrical Projection Mosaic

5.2.10.4 Camera Point Perspective Mosaic

MIPL creates the Camera Point Perspective by using the output camera model (described by the GEOMETRIC_CAMERA_MODEL group in the output mosaic) to project the pixel into space. The origin of the view ray is thus the C point of the camera model, with the ray's direction being determined by the camera model. See Section 5.2.3 and references [Ref 8] through [Ref 17] for the mathematics.

Figure 5.2.11.2 shows a Camera Point Perspective mosaic. It is a perspective projection with horizontal epipolar lines. The mosaic behaves as though the "camera" which acquired the image frames was an instrument with a much larger field of view. For PHX, this type of mosaic is in the Payload Frame and is tilted to reflect the position of the Lander relative to the horizon.

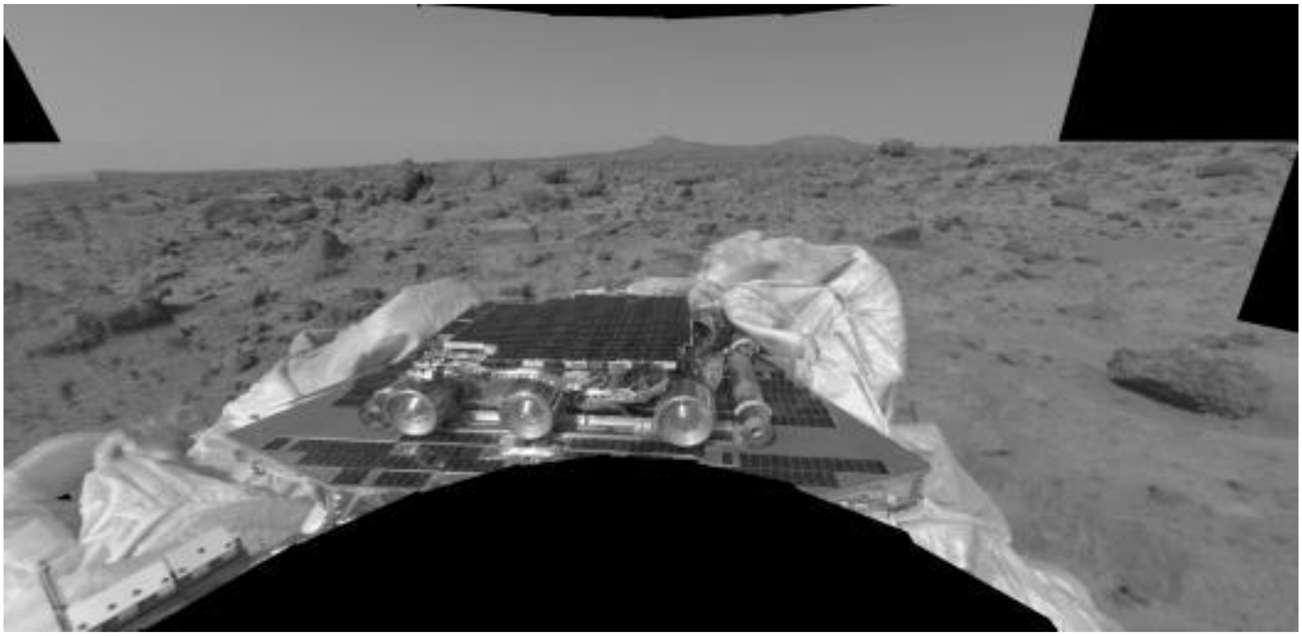


Figure 5.2.11.2 – Camera Point Perspective Mosaic

5.2.10.5 Cylindrical- Perspective Projection Mosaic

At MIPL, this is the most complicated projection to create. Each column i in the output mosaic is assigned its own camera model. This is done in several steps:

1) Compute initial camera model. This model is a CAHV linearized model derived from the first input to the mosaic, and is described in the GEOMETRIC_CAMERA_MODEL label group.

2) The instantaneous field of view of the "central" pixel (at the point where the A vector intersects the image plane) is computed using the formula:

   $$ifov = atan(1.0 / | (\vec{H} - \vec{A} * (\vec{H} \bullet \vec{A})) |)$$

   where the "•" indicates the scalar dot product of the two vectors A and H.

   Alternatively, this can be derived from the image size and azimuthal extent (where the azimuths are adjusted by 360 degrees such that the result is minimally positive):

   ifov = (STOP_AZIMUTH - START_AZIMUTH) / LINE_SAMPLES

3) The azimuth of the column is computed:

   azimuth = START_AZIMUTH + i * ifov

4) The initial camera model is re-pointed using kinematics as described above under the pointing correction section, using the above azimuth and PROJECTION_ELEVATION. This results in the final camera model for the column.

Note that for the PHX masted cameras (SSI), the C points of the column camera models describe a ring in space, whose diameter is approximately the baseline between the cameras, whose plane is approximately horizontal in the Payload Frame, and whose origin is at PROJECTION_ORIGIN_VECTOR. Unfortunately, the MIPL software does not fill in PROJECTION_ORIGIN_VECTOR, thus some information is missing unless knowledge of the kinematics is available. The PROJECTION_ORIGIN_VECTOR may be approximately reconstructed by moving the C point of the initial camera model in a direction normal to the A vector, horizontal in the Lander frame (no Lander frame Z component), and toward the center of the mast (to the right for a left eye, to the left for a right eye, if you are sitting at C and looking along A). Move it by a distance equal to half of the stereo pair's baseline (see Section 2.1: 7.5 cm for SSI). This is an approximation but should be correct to within a few centimeters. [

Once the camera models have been defined, the mosaic proceeds through each pixel as with the other projections. The view ray is computed as described below (A, H, and V come from the column's camera model):

$$x\_center = \vec{A} \cdot \vec{H}$$
$$y\_center = \vec{A} \cdot \vec{V}$$
samp = x_center
line = y_center + j - PROJECTION_ELEVATION_LINE

where the "•" indicates the scalar dot product of two vectors. This (samp,line) coordinate is then projected into space using the column's camera model, and this projection becomes the view ray. The origin of the view ray is the column's C point. See Section 5.2.3 and references [Ref 8] through [Ref 17] for the mathematics of camera models.

Figure 5.2.11.3 shows a Cylindrical-Perspective projection in which a 360 degree view can be viewed in stereo. This is a perspective projection similar to Figure 5.2.11.2 except that the mosaic acts like a pinhole camera which follows the mosaic in azimuth. If the mosaic is generated using Lander coordinates, the horizon will not be level, instead being sinusoidal. This preserves epipolar alignment and allows for better stereo viewing of the panorama. However, for aesthetic reasons, many Cylindrical-Perspective mosaics are created using landing site coordinates. In these cases, the horizon will be level, but stereo alignment will be compromised. If the lander is sitting relatively level, the stereo misalignment is small enough that it does not affect normal viewing, so the aesthetic impact of a flat horizon is often considered more important. The REFERENCE_COORD_SYSTEM_NAME label in SURFACE_PROJECTION_PARMS indicates the frame used.

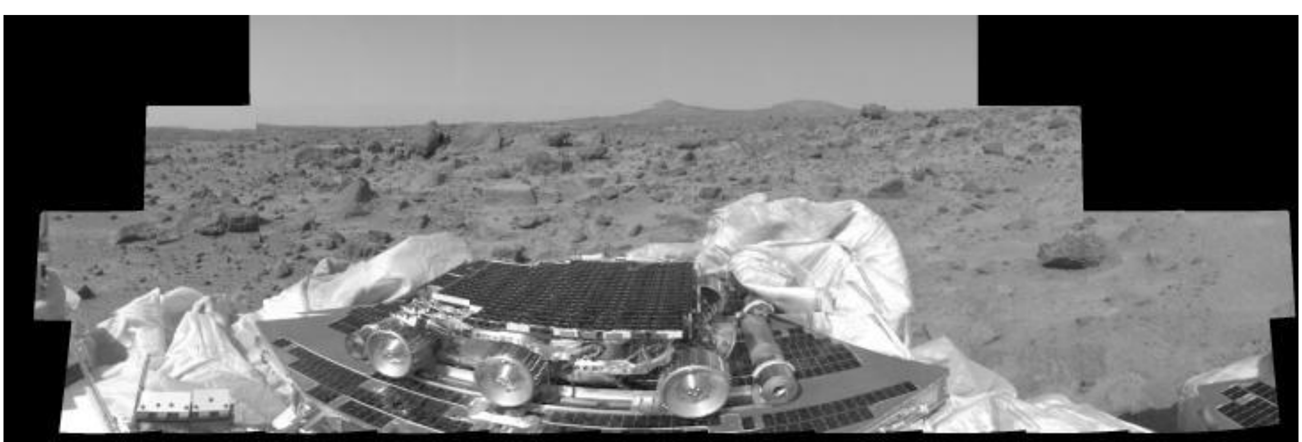


Figure 5.2.11.3 – Cylindrical-Perspective Projection Mosaic

5.2.10.6 Polar Projection Mosaic

MIPL creates the Polar projection by computing the azimuth and elevation of the view ray as follows:

x = i - SAMPLE_PROJECTION_OFFSET
y = LINE_PROJECTION_OFFSET - j
range = sqrt(x*x + y*y)
elevation = range / MAP_RESOLUTION - 90 degrees
azimuth = REFERENCE_AZIMUTH + (90 degrees - atan2(y, x)) / MAP_RESOLUTION

The view ray emanates from the point PROJECTION_ORIGIN_VECTOR.

Figure 5.2.11.4 shows a Polar projection. Concentric circles represent constant projected elevation. Mars nadir is at the convergent center and the horizon is corrected for lander tilt. North is up.

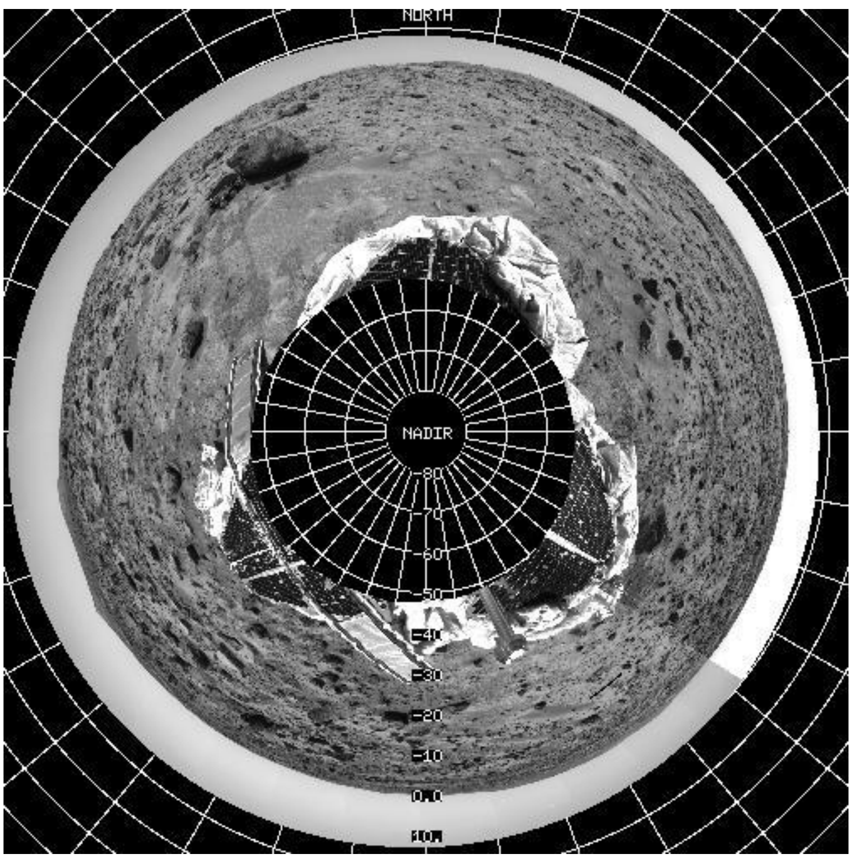


Figure 5.2.11.4 – Polar Projection Mosaic

5.2.10.7 Vertical Projection Mosaic

MIPL creates the Vertical projection as follows:

nl = number of lines in the mosaic (IMAGE object, LINES)
ns = number of samples in the mosaic (IMAGE object, LINE_SAMPLES)
x = (nl/2 - j) * MAP_SCALE
y = (i - ns/2) * MAP_SCALE

The view ray emanates from (x, y, 0) and points straight down (0,0,1). If the ray misses the surface in step E of Section 5.2.12.2 above, it is changed to point straight up (0,0,-1).

Figure 5.2.11.5 shows a vertical view. It assumes that the field is a plane tangent to the Martian surface with up pointing north. This is not an orthorectified rendering but was found to be useful for rapid initial orientation.

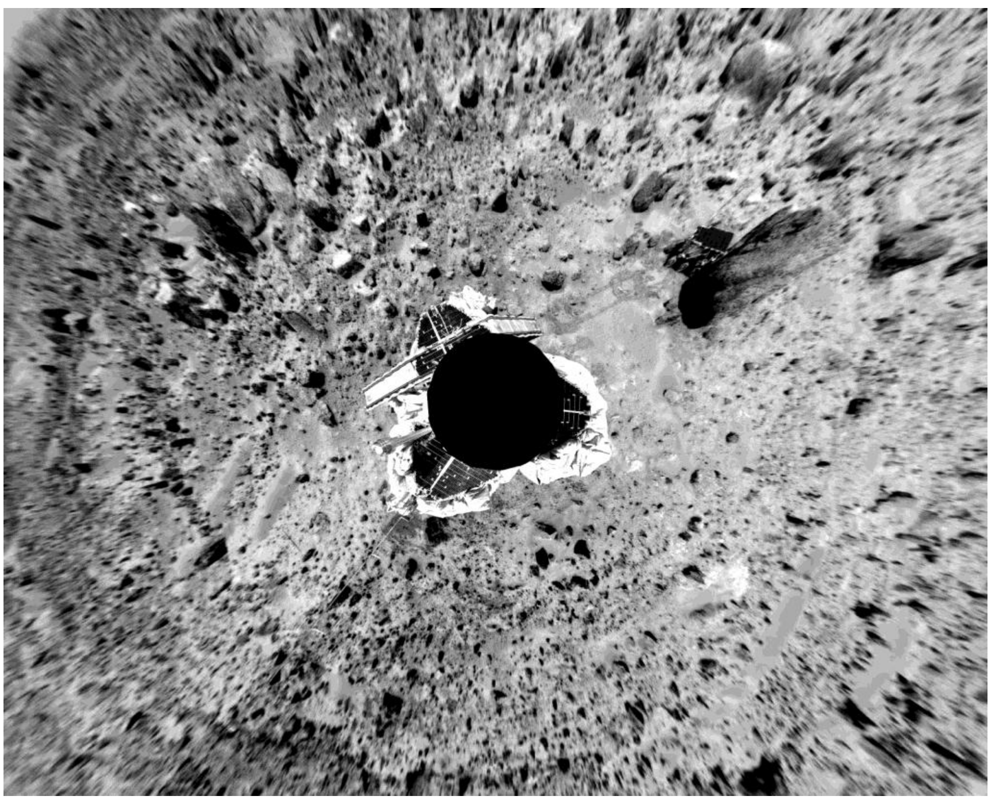


Figure 5.2.11.5 – Vertical Projection Mosaic

5.2.10.8 XYZ Mosaic

Normally mosaics are created using imagery, where each pixel is either a raw or radiometrically corrected intensity value. However, mosaics can also be created using other types of pixels.

An XYZ mosaic contains XYZ values for each pixel in the mosaic rather than intensity values. The inputs to the mosaic program are XYZ files (or individual X, Y, or Z components), and the pixels are interpreted in the same way - as the coordinate of the corresponding pixel in Cartesian space.

Like XYZ images, they may consist of a single 3-band file with X, Y, and Z components, or separate 1 band files for each component.

XYZ mosaics can be produced in any of the mosaic projections.

Care must be taken while producing these mosaics to ensure that a consistent coordinate system is used for all the input images. The output mosaic may have only one coordinate system in which the XYZ values are defined.

5.2.10.9 Surface Normal (UVW) Mosaic

Similar in concept to XYZ mosaics, a UVW mosaic is simply a mosaic created from UVW (surface normal) input images. The pixels represent the surface normals at each point. Like Surface Normal (UVW) images, they can be single 3-band files or separate 1-band files for each component.

As with XYZ mosaics, any projection may be used, and all output values must be defined in the same coordinate system.

5.2.11 Anaglyph RDR

A stereo anaglyph is a method of displaying stereo imagery quickly and conveniently using conventional display technology (no special hardware) and red/blue glasses. This is done by displaying the left eye of the stereo pair in the red channel, and displaying the right eye in the green and blue channels. An anaglyph data product simply captures that into a single 3-band color image, which can be displayed using any standard image display program with no knowledge that it is a stereo image. The red (first) band contains the left eye image, while the green and blue (second and third) bands each contain the right eye image (so the right image is duplicated in the file).

The Anaglyph method can also apply to multi-frame mosaic products. MIPL-generated mosaic Anaglyphs occasionally required some subtle pixel-shifting of the right eye mosaic data to improve the stereo effects. Mosaic Anaglyph products are distinguishable in the Mosaic RDR filename convention (see Section 4.4.2).

6. STANDARDS USED IN GENERATING PRODUCTS

6.1 PDS Standards

The PHX camera instrument EDR data product complies with Planetary Data System standards for file formats and labels, as specified in the PDS Standards Reference [Ref 3]. See Section 4.2 for a description of the PDS Label and the specific conventions adopted by PHX.

6.2 Time Standards

The EDR PDS label uses keywords containing time values. Each time value standard is defined according to the keyword description. See Appendix B.

6.3 Coordinate Frame Standards

This section describes the primary coordinate systems defined for PHX surface operations, which are listed in Table 6.3 and illustrated in Figure 6.3.1 below.

Table 6.3 - Coordinate Frames Used for PHX Surface Operations

Imaging-Related Coordinate Frames		Coordinate Frame Origin	Coordinate Frame Orientation
Name	**Label Keyword Value**		
Payload Frame (P Frame) (RA Frame)	"PAYLOAD_FRAME"	Attached to Lander at intersection of RA torso joint rotation axis	Fixed offset frame rotated 180 deg about +Z axis relative to Lander frame: • +Z axis is normal to deck surface and points from that surface downward. • +X axis is perpendicular to Z axis and points towards RA side of deck. • +Y completes the right-handed frame.
Lander Frame (L Frame) NOTE: Not used in PHX Surface Operations	does not appear in label	Centered on launch vehicle separation plane 940.8 mm above Lander deck	Aligned with Lander: • +Z axis is normal to deck surface and points from that surface downward. • +X axis is perpendicular to Z axis, parallel to solar array yoke and points toward deck side opposite to SSI. • +Y completes the right-handed frame.
SSI Frame (S_{SSI} Frame)	does not appear in label	Attached to Camera	Aligned with camera pointing
Site (S_N Frame) (Surface Frame)	"SITE_FRAME"	Attached to Surface	Aligned with Lander: • +Z axis points downward to Nadir (gravity vector). • +X axis is perpendicular to

Imaging-Related Coordinate Frames		Coordinate Frame Origin	Coordinate Frame Orientation
Name	Label Keyword Value		
			Z axis and points towards North. • +Y completes the right-handed frame and points East.
Local Level (L_L Frame)	"LOCAL_LEVEL_FRAME"	Attached to Lander (coincident with Payload Frame)	Fixed offset frame relative to Site frame: • +Z axis points downward to Nadir (gravity vector). • +X axis is perpendicular to Z axis and points towards North. • +Y completes the right-handed frame and points East.

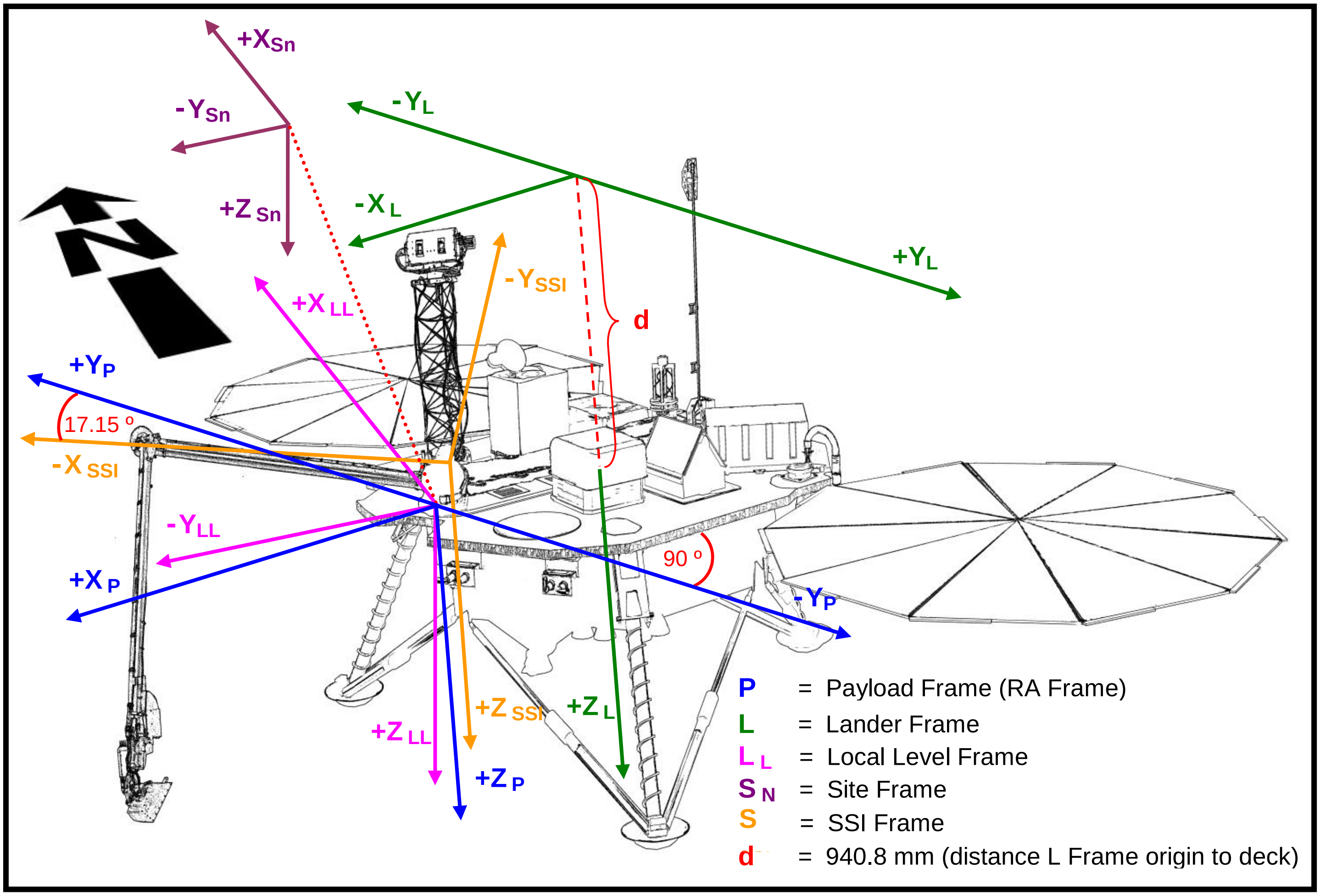


Figure 6.3.1 - L, P, L_L, S_N, and S_{SSI} Coordinate Frames

6.3.1.1 Payload Frame (RA Frame)

The Payload frame is the one used for all surface operations and commanding. It is the only frame understood by the flight software for camera pointing and arm commanding. It is analogous to the MER "Rover" frame. The Payload frame is attached to the lander, and moves with it should the lander move while on the surface. Its origin is defined as the intersection of the RA torso joint rotation axis and the deck surface. It is aligned with the Lander frame, although with a 180 degree rotation about Z. Thus, +Y points in the same direction as Lander -Y (roughly towards the leg closest to the RA), while +X points in the same direction as Lander -X. Both Payload +X and +Y point into the RA workspace. +Z points down. The Payload frame is defined relative to its enclosing Site frame (see below) by an offset and quaternion value (see the PAYLOAD_COORDINATE_SYSTEM label group).

Various other RA Tool frames exist, but do not appear in the label, and are not defined further here.

6.3.1.2 Lander Frame

The Lander frame is not used for any surface operations, and will not appear in the label. For reference, its origin is at X=-0.5336, Y=-0.5341, Z=-0.9408 meters as measured in the Payload frame.

6.3.1.3 SSI Frame

Three SSI frames are used internally, but do not appear in the label. They are defined here for completeness. The SSI frame is defined relative to the Payload frame. It is rotated -107.15 degrees about the payload Z axis (thus SSI +Y points 17.15 degrees from Payload +X), while its origin is the "intersection" of the SSI azimuth and elevation axes, at X=-0.4411, Y=-0.1140, Z=-0.7935 meters as measured in the Payload frame. The SSI Home frame is defined similarly, except with an additional +Z rotation of -341.6 degrees from the SSI frame. SSI Home is used to calculate azimuth and elevation based on motor counts and the backlash tables; azimuth and elevation of 0 in this frame represent motor counts of 0. The SSI Head frame is attached to the camera and is oriented using gimbal angles (motor counts and backlash tables). Its origin is the same intersection of az/el axes, while +X is perpendicular to the elevation axis, in the plane defined by the left eye boresight vector; +Y is parallel to the elevation axis and points from the origin towards the left eye; and +Z completes the right-handed frame.

6.3.1.4 Local Level and Site Frames

Two additional frames are inherited from MER for ground operations: Local_Level and Site. Local_Level is defined to be coincident in origin with Payload at all times (thus it moves with the lander), but is oriented with +X pointing North, +Z pointing down along the local gravity vector, and +Y completing the right-hand frame (thus, it points East). Site frames are oriented parallel to Local_Level. They are coincident in origin at the time the Site is declared, but Sites do not move with any subsequent lander movement (thus, they are "pinned to the ground").

It is important to realize that Local_Level and Site are purely ground constructs: the lander knows nothing of them. They exist to accommodate the possibility of the lander moving after landing, due to settling, arm operations, or other factors. They also make certain kinds of observations more convenient to command; thus ground tools such as PSI, RSVP, and the MIPL tools should be able to convert to and from them. Local_Level and Site are indexed via a "Rover Motion Counter" (RMC) (the name is the result of MER heritage). For Phoenix, the RMC has only two values: Site and Position. Position is used to track small motions of the lander, while a new Site may be declared to zero out those motions for ease of commanding. It is anticipated that there will be no more than one or two Sites declared in the duration of the mission.

The initial orientation of the Local_Level and Site frames (RMC index 0,0) is derived from on-board sensors, but soon after landing these are turned off. Further updates to position and orientation will be derived on the ground via image analysis, by the operations or science teams. The current position and orientation are maintained by MIPL in an "RMC database", which is a set of XML files defining the quaternion and offset of each Position relative to its Site, and the offset of the Sites from each other. Unlike MER, these files also contain the SCLK value for the time at which the RMC value was incremented. This RMC database is consulted when creating EDR's in order to fill in the PAYLOAD_COORDINATE_SYSTEM values.

Thus, the EDR label contains the orientation and offset of the current Payload frame with respect to the current Site. Like MER, site-to-site offsets are not in the label; the RMC database must be consulted for those.

Because site/position information is updated based on image analysis, the EDR's used to do the analysis will have incorrect position information in their labels. For this reason, such EDR's may be re-processed to retroactively reflect the updated position information.

6.3.1.5 MECA-OM Frame

The MECA-OM Frame is defined by the two mechanical degrees of freedom of the MECA sample stage. Rotation of the stage corresponds to a left-right (x-axis) motion of the image in the field, while translation of the stage corresponds to motion of the sample in and out of the focal plane (z-axis). Both are specified by step numbers relative to a designated limit switch (LS). See Figure 6.3.2 below for rotation LS position.

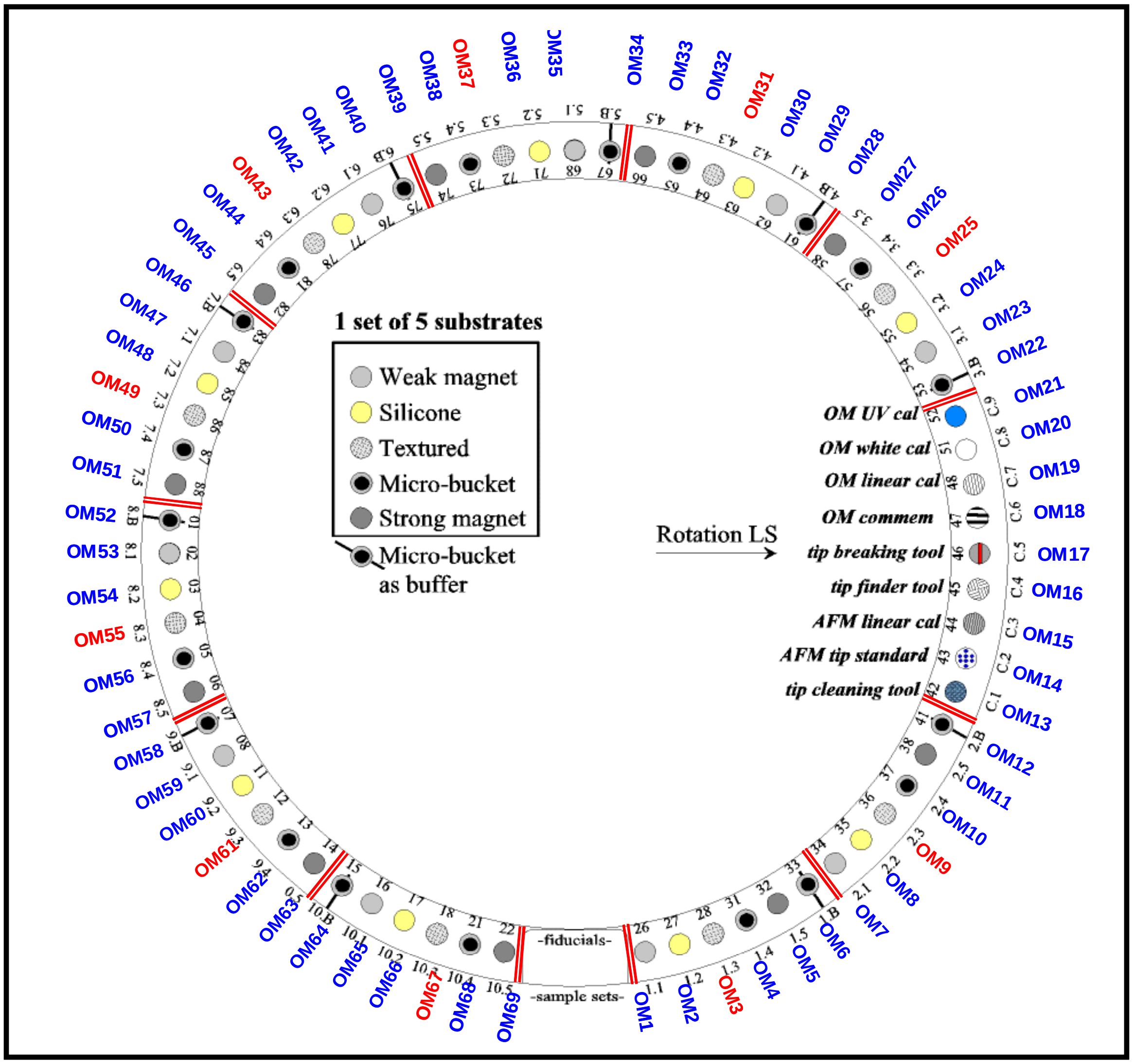


Figure 6.3.2 - MECA-OM Sample Stage Steps

The rotation positions run from 0 to 24000 clockwise from the Rotation LS. Each step corresponds to a rotation of 15 µm. Each substrate is 3 mm in diameter (200 steps) with a 2-mm gap between substrates such that 1000 steps advances by 3 substrates. There are 69 substrates numbered OM1 to OM69 (were the notch not present, there would be 72 substrates for a total perimeter of 360 mm).

Translation is measured relative to a limit switch at the "OUT" position (farthest from the OM), which is arbitrarily designated step 10,000 to allow for possible shifts. The nominal step size is 0.25 µm and typical focus position is on the order of 50,000, corresponding to 10 mm travel inward from "OUT."

7. APPLICABLE SOFTWARE

PHX camera instrument downlink processing software is focused on rapid reduction, calibration, and visualization of images in order to make discoveries, to accurately and expeditiously characterize the geologic environment around the Lander, and to provide timely input for operational decisions concerning RA navigation and target selection. Key software tools have been developed at the University of Arizona, Texas A&M University, at JPL by the MIPL, SSV, and APSS groups. These tools can also be used to process SSI, RAC and MECA-OM images, which have substantial scientific potential in addition to their operational importance

7.1 Utility Programs

Table 7.1 lists (in no particular order) the primary software tools that will be used to process and manipulate downlinked PHX camera instrument imaging data. All image processing software will be executable by members of the three PHX camera instrument teams on computers in the SOC facility at the University of Arizona and will be capable of reading and writing image data in PDS format. The Operations Product Generation System (OPGS) and MIPL will generate EDRs in PDS format and deposit them on their FEI server for transfer to the University of Arizona's PHX GDS filesystem as rapidly as possible after receipt of telemetry.

Table 7.1 - Key Software Tools for PHX Camera Payload Downlink Processing

Name	Description	Primary Development Responsibility
PHXTELEMPROC	Fetches the image Standard Formatted Data Unit (SFDU) records from PHX instrument telemetry packets, reconstructing the image data into a PDS-labelled image EDR data product. VICAR code.	Payam Zamani (JPL / MIPL)
SSI Instrument Team Software	SSITOOLS: Quick-look toolkit for image browsing and display, spectral manipulation, and mosaic generation: • MERSTAMPS – Quick-look image browsing. • MERVIEW – Quick-look image analysis. • SSI_SPEC – Qick-look spectral processing. • SSIMAP2 – Mosaic generation. • TAMCAL – Generates radiometrically-corrected images and I/F-corrected images.	Mark Lemmon (Texas A&M University)
RAC Instrument Team Software	RACSoft: Provides radiometric correction functionality for processing of RAC and MECA-OM data.	Subbe Hviid (MPS)
MECA-OM Instrument Team Software	Same as RAC software tools.	Michael Hecht (JPL)
Mars Program Suite	Rapid ("quick look") correlator-based mosaic generation using raw EDRs or calibrated images from SSI, RAC and/or MECA-OM. VICAR code: • MARSCAHV – Generates a geometrically corrected version of the EDR, applying the C, A, H and V camera model vectors.	Bob Deen (JPL / MIPL)

Name	Description	Primary Development Responsibility
	• MARSRAD – Generates a radiometrically corrected image from a single input EDR. • MARSJPLSTEREO – Generates a disparity map from a stereo pair of input EDRs, applying a 1-D correlator (fast). • MARSCOR3 – Generates a disparity map from a stereo pair of input EDRs, applying a 2-D correlator (more robust). • MARSXYZ – Generates an XYZ image from an input disparity map. • MARSRANGE – Generates an Range image from an input XYZ map. • MARSREACH – Generates an IDD reachability map from an input XYZ map. • MARSUVW - Generates a Surface Normal image, wherein XYZ is computed normal to the surface. • MARSMAP – Generates a Cylindrical, Polar or Vertical projection mosaic from a list of input EDRs. • MARSMOS – Produces pinhole camera mosaics using uncorrected input images and CAHVOR camera model. • MARSMCAULEY – Generates a combination Cylindrical-Perspective projection mosaic from a list of input EDRs. • MARSTIE – Generates pointing corrections (tiepoint file) from an overlapping set of input EDRs. • MARSNAV – Generates an updated azimuth and elevation file based on comparison with existing image data that can be directly compared. • MICA – Interactive mosaic correction.	
SUMMITT	Performs stereo processing to create XYZ images and 3D polygon mesh terrain models.: • do_wedge – A script tool that calls lower-level code to generate terrain triangles, or “wedges”, and created a “mesh” for each input XYZ file. • merge_mesh – A script tool that calls lower-level code to merge individual terrain meshes into a single “unified” surface fitted terrain mesh in Performer Binary format suitable for ingestion by RSVP software.	John Wright (JPL / MIPL)
APSS / PSI	Visualization and planning software for creation of science products and candidate observations for presentation at Ops planning meetings, and then Sol activity list at end of planning meetings. Java code.	Jason Fox (JPL)
APSS / RSVP	Visualization, planning, and sequence generation software for use by Sequence Team to create Sol sequences based on activity lists generated by PSI during planning meetings. Java, C and C++ code.	Brian Cooper (JPL)

7.2 Applicable PDS Software Tools

PDS-labeled images and tables can be viewed with the program NASAView, developed by the PDS and available for a variety of computer platforms from the PDS web site http://pds.jpl.nasa.gov/tools. There is no charge for NASAView.

7.3 Software Distribution and Update Procedures

The FEI distribution tool and Mars Image Processing Program Suite are available to researchers and academic institutions. Refer to the MIPL Web site at http://www-mipl.jpl.nasa.gov for contact information. FEI is described in detail at http://www-mipl.jpl.nasa.gov/MDMS.html.

APPENDIX A – Camera EDR & RDR Label Keyword Definitions

Phoenix uses a dual-label strategy. Each product contains both a PDS and VICAR labels. Where they differ and applicable, VICAR keywords are identified with a superscript "V". (^{V})

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
APPLICATION_PROCESS_ID	Specifies the process, or source, which created the data.	APID — INSTRUMENT "49" to "64" — MECA "69" to "74" — RA "75" to "82" — RAC "83" to "95" — SSI	**SOURCE** apid **LOCATION** TELEMETRY (Class)	**TYPE** integer **UNITS** n/a
APPLICATION_PROCESS_NAME	Specifies the name associated with the source or process which created the data.	NAME — APID "APID_MECA_AFM_CRIT" 49 "APID_MECA_AFM_KEY_STRAT" 50 "APID_MECA_AFM_KEY_TACT" 51 "APID_MECA_ANCIL_KEY_STRAT" 52 "APID_MECA_ANCIL_KEY_TACT" 53 "APID_MECA_ANCIL_KEY_TACT_CRITICAL" 54 "APID_MECA_NOMINAL" 55 "APID_MECA_OM_CRIT" 56 "APID_MECA_OM_KEY_STRAT" 57 "APID_MECA_OM_KEY_TACT" 58 "APID_MECA_TECP_CRIT" 59 "APID_MECA_TECP_KEY_STRAT" 60 "APID_MECA_TECP_KEY_TACT" 61 "APID_MECA_WC_CRIT" 62 "APID_MECA_WC_KEY_STRAT" 63 "APID_MECA_WC_KEY_TACT" 64 "APID_RA_COL_DB" 69 "APID_RA_HIST" 70 "APID_RA_PARAMS" 71 "APID_RA_SCI_HI_PRI" 72 "APID_RA_SCI_LO_PRI" 73 "APID_RA_STATUS" 74 "APID_RAC_CRIT_HI" 75 "APID_RAC_CRIT_KEY" 76	**SOURCE** Table Lookup: • apid **LOCATION** TELEMETRY (Class)	**TYPE** string(256) **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"APID_RAC_CRIT_LOW" 77 "APID_RAC_CRIT_MED" 78 "APID_RAC_HI" 79 "APID_RAC_KEY" 80 "APID_RAC_LOW" 81 "APID_RAC_MED" 82 "APID_SSI_CRIT_HI_1" 83 "APID_SSI_CRIT_HI_2" 84 "APID_SSI_CRIT_KEY_1" 85 "APID_SSI_CRIT_KEY_2" 86 "APID_SSI_CRIT_LOW" 87 "APID_SSI_CRIT_MED" 88 "APID_SSI_HI_1" 89 "APID_SSI_HI_2" 90 "APID_SSI_KEY_1" 91 "APID_SSI_KEY_2" 92 "APID_SSI_LOW" 93 "APID_SSI_MED_1" 94 "APID_SSI_MED_2" 95		
ARTICULATION_DEVICE_ANGLE	Specifies the value of an angle between two parts or segments of an articulated device. For SSI instrument articulation, array elements: 1) Azimuth position at time of exposure adjusted for backlash (radians). 2) Elevation position at time of exposure adjusted for backlash (radians). 3) Azimuth position at time of exposure (radians). 4) Elevation position at time of exposure (radians). 5) Azimuth position before move (radians). 6) Elevation position before move (radians). For RA instrument articulation, array elements: 1) RA joint encoder angle - joint 1 (radians). 2) RA joint encoder angle - joint 2 (radians). 3) RA joint encoder angle - joint 3 (radians). 4) RA joint encoder angle - joint 4 (radians).	SSI Array elements: 1) "0.0" to "<2π>" 2) "<-π/2>" to "<π/2>" 3) "0.0" to "<2π>" 4) "<-π/2>" to "<π/2>" 5) "0.0" to "<2π>" 6) "<-π/2>" to "<π/2>" where π = 3.1415926535897932384 RA Array elements: 1) "-3.3" to "1.8" 2) "-2.5" to "1.7" 3) "-0.9" to "4.1" 4) "-0.2" to "3.7"	**SOURCE** Group Dependent, calculations by instrument: a) For SSI, array elements: 1) 0.0 + (az_pos_curr * 0.288) 2) el_pos_curr * 0.288 3) 0.0 + (az_pos_prev * 0.288) 4) el_pos_prev * 0.288 b) For RA, array elements: 1) ra_enc_1 2) ra_enc_2 3) ra_enc_3 4) ra_enc_4 NOTES: • SSI - For Azimuth, 0.0 is a temporary placeholder for the offset until it can be updated (hard stops are symmetric around ±180 deg). - For Elevation, put in a "-" to make negative values be down (that is, +312 is zenith). 0 deg is parallel to the lander deck, in theory. **LOCATION** Group Dependent: a) SSI_ARTICULATION_STATE (Group) b) RA_ARTICULATION_STATE (Group)	**TYPE** float array[10] **UNITS** radians (<rad> unit tag required)
ARTICULATION_DEVICE_ANGLE_NAME	Specifies the formal name which identifies each of the values used in	SSI Array elements:	**SOURCE** Group Dependent:	**TYPE** string

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	ARTICULATION_DEVICE_ANGLE.	1) "AZIMUTH-BACKLASH" 2) "ELEVATION-BACKLASH" 1) "AZIMUTH-MEASURED" 2) "ELEVATION-MEASURED" 3) "AZIMUTH-INITIAL" 4) "ELEVATION-INITIAL" <u>RA</u> Array elements: 1) "JOINT 1 AZIMUTH" 2) "JOINT 2 ELEVATION" 3) "JOINT 3 ELBOW" 4) "JOINT 4 WRIST"	a) For SSI, array elements: 1) az_pos_curr 2) el_pos_curr 3) az_pos_prev 4) el_pos_prev b) For RA, array elements: 1) ra_enc_1 2) ra_enc_2 3) ra_enc_3 4) ra_enc_4 **LOCATION** Group Dependent: a) SSI_ARTICULATION_STATE (Group) b) RA_ARTICULATION_STATE (Group)	array[10] **UNITS** n/a
ARTICULATION_DEV_MOTOR_CLICKS ARTICULATION_DEVICE_COUNT [V]	Specifies the position of some articulated device expressed in an instrument-specific raw telemetered value (usually motor counts). As such, no unit applies. The information may also be given in other keywords (such as ARTICULATION_DEVICE_ANGLE) expressed in more meaningful units, although that is not required. For SSI instrument: a) For filter articulation, array elements: 1) Current filter wheel position (steps). 2) Starting filter wheel position (steps). b) For instrument articulation, array elements: 1) Azimuth position at time of exposure (motor step counts). 2) Elevation position at time of exposure (motor step counts). 3) Azimuth position before move (motor step counts). 4) Elevation position before move (motor step counts). For RAC instrument articulation, array elements: 1) Focus position at time of exposure. 2) Starting focus position. For RA instrument articulation, array elements: 1) RA tool position in RA frame, X axis (meters).	<u>SSI Filter</u> Array elements: 1) "0" to "359" 2) "0" to "359" <u>SSI</u> Array elements: 1) "-618" to "618" 2) "-312" to "312" 3) "-618" to "618" 4) "-312" to "312" <u>RAC</u> Array elements: 1) "0" to "313" 2) "0" to "313" <u>MECA-OM</u> Array elements: 1) "0" to "65535" 2) "0" to "65535"	**SOURCE** Group Dependent: a) For SSI Filter, array elements: 1) fw_pos_steps_curr 2) fw_pos_steps_prev b) For SSI, array elements: 1) az_pos_curr 2) el_pos_curr 3) az_pos_prev 4) el_pos_prev c) For RAC, array elements: 1) foc_pos_curr 2) foc_pos_prev d) For OM, array elements: 1) translation_pos 2) rotation_pos **LOCATION** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group) b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	**TYPE** integer array **UNITS** steps

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	2) RA tool position in RA frame, Y axis (meters). 3) RA tool position in RA frame, Z axis (meters). 4) RA tool orientation in RA frame, axis 1 (meters). 5) RA tool orientation in RA frame, axis 2 (meters). 6) RA tool orientation in RA frame, axis 3 (meters). 7) RA tool orientation in RA frame, axis 4 (meters). For MECA-OM instrument articulation, array elements: 1) Translation stage position for MECA-OM. 2) Rotation position for MECA-OM.			
ARTICULATION_DEV_MTR_CLCK_NAME ARTICULATION_DEVICE_COUNT_NAME [V]	specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_COUNT." GENERAL_DATA_TYPE = CHARACTER MAXIMUM = "N/A" MINIMUM = "N/A" MAXIMUM_LENGTH = "N/A" MINIMUM_LENGTH = "N/A" STANDARD_VALUE_TYPE = TEXT STANDARD_VALUE_SET_DESC = "N/A" KEYWORD_DEFAULT_VALUE = "N/A" UNIT_ID = "N/A" FORMATION_RULE_DESC = "N/A"	<u>SSI Filter</u> Array elements: 1) "FILTER_POS-CURRENT" 2) "FILTER_POS-PREVIOUS" <u>SSI</u> Array elements: 1) "AZIMUTH-MEASURED" 2) "ELEVATION-MEASURED" 3) "AZIMUTH-INITIAL" 4) "ELEVATION-INITIAL" <u>RAC</u> Array elements: 1) "FOCUS_POS-CURRENT" 2) "FOCUS_POS-PREVIOUS" <u>MECA-OM</u> Array elements: 1) "TRANSLATION_STAGE_POS" 2) "ROTATION_POS"	**<u>SOURCE</u>** Group Dependent: a) For SSI Filter, array elements: 1) fw_pos_steps_curr 2) fw_pos_steps_prev b) For SSI, array elements: 1) az_pos_curr 2) el_pos_curr 3) az_pos_prev 4) el_pos_prev c) For RAC, array elements: 1) foc_pos_curr 2) foc_pos_prev d) For OM, array elements: 1) translation_pos 2) rotation_pos **<u>LOCATION</u>** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group) b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	**<u>TYPE</u>** string array **<u>UNITS</u>** N/A
ARTICULATION_DEVICE_ID	Specifies the unique abbreviated identification of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g., mast heads, wheel bogies, arms, etc.). Note: The ARTICULATION_DEVICE_ID is	"SSI FILTER", "SSI", "RAC", MECA-OM"	**<u>SOURCE</u>** Group Dependent: • Static values **<u>LOCATION</u>** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group)	**<u>TYPE</u>** string **<u>UNITS</u>** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	not a unique identifier for a given articulated device. Note also that the associated ARTICULATION_DEVICE_NAME element provides the full name of the articulated device.		b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	
ARTICULATION_DEVICE_MODE	Specifies the deployment state (i.e., physical configuration) of an articulation device at the time of data acquisition.	SSI "STOWED", "DEPLOYED"	**SOURCE** Table Lookup **LOCATION** SSI_ARTICULATION_STATE (Group)	**TYPE** string **UNITS** n/a
ARTICULATION_DEVICE_NAME	Specifies the common name of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g. mast heads, wheel bogies, arms, etc.) Note: The associated ARTICULATION_DEVICE_ID element provides an abbreviated name or acronym for the articulated device.	"SOLID STATE IMAGER FILTER", "SOLID STATE IMAGER", "ROBOTIC ARM CAMERA", "MECA-OPTICAL MICROSCOPE"	**SOURCE** Group Dependent: • Static values **LOCATION** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group) b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	**TYPE** string **UNITS** n/a
ARTICULATION_DEVICE_COMP_STATE ARTICULATION_DEVICE_STATE [V]	Specifies some set of states in instrument was in when the image was acquired. Each element can be a Boolean (on/off) or a multi-valued state, and valid values vary per instrument. The associated name (in ARTICULATION_DEVICE_STATE_NAME) specifies what component is being described; the value in ARTICULATION_DEVICE_STATE indicates the state of that component. For SSI instrument, indicates the direction in which the mast was last moved in order to reach its current position. Useful for calculating backlash corrections: a) For filter articulation, previous filter wheel movement direction. b) For instrument articulation, array elements: 1) Azimuth movement direction. 2) Elevation movement direction. For RAC instrument articulation, array elements: 1) RAC cover state. 2) RAC Red lamp state. 3) RAC Green lamp state. 4) RAC Blue lamp state.	SSI Filter Array elements: 1) 0 = "CLOCKWISE" 2) 1 = "COUNTERCLOCKWISE" SSI Array elements: 1) 0 = "CLOCKWISE", 1 = "COUNTERCLOCKWISE" 2) 0 = "UP", 1 = "DOWN" RAC Array elements: 1) 0 = "OPEN", 1 = "CLOSED" 2) "ON", "OFF" 3) "ON", "OFF" 4) "ON", "OFF" OM All 16 array elements: "ON", "OFF"	**SOURCE** Group Dependent: a) fw_dir b) az_dir, el_dir c) cover_status, lamp_state d) led_bitmask NOTE: Lookup table is used with lamp_state for proper conversions. **LOCATION** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group) b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	**TYPE** string array **UNITS** N/A

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	For OM instrument articulation, all 16 array elements: - "ON/OFF" states for 4 sets of LED's (RED, GRN, BLU, UV per set) derived from LED state bitmask. LEDs can be in combination.			
ARTICULATION_DEVICE_COMP_NAME ARTICULATION_DEVICE_STATE_NAME [V]	Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_STATE.	SSI Filter Array elements: 1) "FILTER_DIR" SSI Array elements: 1) "AZ_DIR" 2) "EL_DIR" RAC Array elements: 1) "RAC_COVER" 2) "RAC_LAMP_RED" 3) "RAC_LAMP_GRN" 4) "RAC_LAMP_BLU" OM 1st thru 16th element (left to right, top to bottom): "R1", "G1", "B1", "UV1", "R2", "G2", "B2", "UV2", "R3", "G3", "B3", "UV3", "R4", "G4", "B4", "UV4"	**SOURCE** Group Dependent: a) fw_dir b) az_dir, el_dir c) cover_status, lamp_name d) led_bitmask e) led_bitmask **LOCATION** Group Dependent: a) SSI_FILTER_ARTICULATION_STATE (Group) b) SSI_ARTICULATION_STATE (Group) c) RAC_ARTICULATION_STATE (Group) d) OM_ARTICULATION_STATE (Group)	**TYPE** string array **UNITS** N/A
ARTICULATION_DEV_LOCATION	Specifies the location, in XYZ space, of one or more identified points on a multi-jointed arm or similar device mounted on a spacecraft. The coordinate system in which the values are expressed must be identified in the label either via the GROUP in which this keyword is included (e.g. Phoenix) or through an associated keyword like COORDINATE_SYSTEM_NAME. The number of values must be a multiple of 3 (XYZ coordinates for any number positions). See also ARTICULATION_DEV_LOCATION_NAME. NOTE: For Phoenix, in the case of the Robotic Arm (RA) instrument articulation, this keyword indicates the location of the RA tool expressed in meters in the Payload Frame (which differs from the Lander	RA Array elements: 1) "-2.7" to "2.7" 2) "-2.7" to "2.7" 3) "-2.7" to "2.7"	**SOURCE** For RA, array elements: 1) ra_pos_x 2) ra_pos_y 3) ra_pos_z **LOCATION** RA_ARTICULATION_STATE (Group)	**TYPE** float array **UNITS** meters

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	frame) associated with the image.”.			
ARTICULATION_DEV_LOCATION_NAME	Provides the formal names of one or more identified points on a multi-jointed arm or similar device mounted on a spacecraft, whose locations are specified using the ARTICULATION_DEV_LOCATION keyword. The number of values for this keyword will have a 1:3 correspondence with the number of values of the ARTICULATION_DEV_LOCATION keyword. E.g., for an articulation device with two points identified, there will be two names but six coordinate values See also ARTICULATION_DEV_LOCATION.	<u>RA</u> Array elements: 1) “RA TOOL POSITION - X” 2) “RA TOOL POSITION - Y” 3) “RA TOOL POSITION - Z”	**<u>SOURCE</u>** Static value **<u>LOCATION</u>** RA_ARTICULATION_STATE (Group)	**<u>TYPE</u>** string array **<u>UNITS</u>** N/A
ARTICULATION_DEV_ORIENT_QUAT ARTICULATION_DEV_ORIENT [V]	`specifies the orientation of one or more elements (named by ARTICULATION_DEV_LOCATION_NAME) as a quaternion. The quaternion is expressed in the order (s, v1, v2, v3). See ORIGIN_ROTATION_QUATERNION for further definition. The coordinate system in which the orientations are expressed is implied by the group in which the keyword appears. The number of values must be multiple of 4.For PHX, in the case of RA instrument articulation, this keyword indicates the orientation of the RA tool expressed in meters in the Payload Frame (different from Lander frame) associated with the image. See also ARTICULATION_DEV_ORIENT_NAME."` `GENERAL_DATA_TYPE = REAL` `MAXIMUM = 1.0` `MINIMUM = -1.0` `MAXIMUM_LENGTH = "N/A"` `MINIMUM_LENGTH = "N/A"` `STANDARD_VALUE_TYPE = RANGE` `STANDARD_VALUE_SET_DESC = "N/A"` `KEYWORD_DEFAULT_VALUE =`	<u>RA</u> Array elements: 1) “-1.0” to “1.0” 2) “-1.0” to “1.0” 3) “-1.0” to “1.0” 4) “-1.0” to “1.0”	**<u>SOURCE</u>** For RA, array elements: 1) ra_orient_1 2) ra_orient_2 3) ra_orient_3 4) ra_orient_4 **<u>LOCATION</u>** RA_ARTICULATION_STATE (Group)	**<u>TYPE</u>** float array **<u>UNITS</u>** N/A

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	"N/A" UNIT_ID = "N/A" FORMATION_RULE_DESC = "N/A" See also ARTICULATION_DEV_ORIENT_NAME.			
ARTICULATION_DEV_ORIENT_NAME	Provides the formal names of one or more identified points on a multi-jointed arm or similar device mounted on a spacecraft, whose orientations are specified using the ARTICULATION_DEV_ORIENT keyword. The number of values for this keyword will have a 1:4 correspondence with the number of values of the ARTICULATION_DEV_ORIENT keyword. E.g., for an articulation device with two points identified, there will be two names but eight values. See also ARTICULATION_DEV_ORIENT.	RA Array elements: 1) “RA TOOL ORIENT - 1” 2) “RA TOOL ORIENT - 2” 3) “RA TOOL ORIENT - 3” 4) “RA TOOL ORIENT - 4”	**SOURCE** For RA, array elements: 1) ra_orient_1 2) ra_orient_2 3) ra_orient_3 4) ra_orient_4 **LOCATION** RA_ARTICULATION_STATE (Group)	**TYPE** string array **UNITS** N/A
ARTICULATION_DEV_POSITION	Specifies the set of indices for articulation devices that contain moving parts with discrete positions. The associated ARTICULATION_DEV_POSITION_NAME element names each moving device, and ARTICULATION_DEV_POSITION_ID provides a textual identifier that maps to the position indices. See also ARTICULATION_DEV_POSITION_ID.	SSI (Flight Model) • Left Eye: “1” = SSI_L1_672NM “2” = SSI_L2_445NM “3” = SSI_L3_451NM “4” = SSI_L4_990NM “5” = SSI_L5_886NM “6” = SSI_L6_830NM “7” = SSI_L7_802NM “8” = SSI_L8_861NM “9” = SSI_L9_901NM “10” = SSI_L10_932NM “11” = SSI_L11_1001NM “12” = SSI_L12_967NM • Right Eye: “1” = SSI_R1_672NM “2” = SSI_R2_445NM “3” = SSI_R3_671NM “4” = SSI_R4_936NM “5” = SSI_R5_936NM “6” = SSI_R6_445NM “7” = SSI_R7_753NM “8” = SSI_R8_753NM “9” = SSI_R9_753NM “10” = SSI_R10_604NM “11” = SSI_R11_533NM “12” = SSI_R12_485NM SSI (Engineering Model)	**SOURCE** Table lookup **LOCATION** SSI_ARTICULATION_STATE (Group) OM_ARTICULATION_STATE (Group)	**TYPE** integer array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		• Left Eye: “1” = SSI_L1_445NM “2” = SSI_L2_451NM “3” = SSI_L3_886NM “4” = SSI_L4_OPEN “5” = SSI_L5_935NM “6” = SSI_L6_672NM “7” = SSI_L7_802NM “8” = SSI_L8_D_OPEN “9” = SSI_L9_901NM “10” = SSI_L10_932NM “11” = SSI_L11_1001NM “12” = SSI_L12_967NM • Right Eye: “1” = SSI_R1_446NM “2” = SSI_R2_671NM “3” = SSI_R3_935NM “4” = SSI_R4_OPEN “5” = SSI_R5_990NM “6” = SSI_R6_672NM “7” = SSI_R7_753NM “8” = SSI_R8_D_OPEN “9” = SSI_R9_604NM “10” = SSI_R10_533NM “11” = SSI_R11_485NM “12” = SSI_R12_966NM <u>MECA-OM Stage Rotation Position</u> Step Range: Substrate Range: “0” – “999” OM1 – OM3 “1000” – “1999” OM4 – OM6 “2000” – “2999” OM7 – OM9 “3000” – “3999” OM10 – OM12 “4000” – “4999” OM13 – OM15 “5000” – “5999” OM16 – OM18 “6000” – “6999” OM19 – OM21 “7000” – “7999” OM22 – OM24 “8000” – “8999” OM25 – OM27 “9000” – “9999” OM28 – OM30 “10000” – “10999” OM31 – OM33 “11000” – “11999” OM34 – OM36 “12000” – “12999” OM37 – OM39 “13000” – “13999” OM40 – OM42 “14000” – “14999” OM43 – OM45 “15000” – “15999” OM46 – OM48 “16000” – “16999” OM49 – OM51 “17000” – “17999” OM52 – OM54		

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"18000" – "18999" OM55 – OM57 "19000" – "19999" OM58 – OM60 "20000" – "20999" OM61 – OM63 "21000" – "21999" OM64 – OM66 "22000" – "22999" OM67 – OM69		
ARTICULATION_DEV_POSITION_ID	Specifies the set of identifiers corresponding to ARTICULATION_DEV_POSITION. These describe the position (e.g. filter), not the device (e.g., filter wheel). See also ARTICULATION_DEV_POSITION.	SSI (Flight Model) • Left Eye: 1 = "SSI_L1_672NM" 2 = "SSI_L2_445NM" 3 = "SSI_L3_451NM" 4 = "SSI_L4_990NM" 5 = "SSI_L5_886NM" 6 = "SSI_L6_830NM" 7 = "SSI_L7_802NM" 8 = "SSI_L8_861NM" 9 = "SSI_L9_901NM" 10 = "SSI_L10_932NM" 11 = "SSI_L11_1001NM" 12 = "SSI_L12_967NM" • Right Eye: 1 = "SSI_R1_672NM" 2 = "SSI_R2_445NM" 3 = "SSI_R3_671NM" 4 = "SSI_R4_936NM" 5 = "SSI_R5_936NM" 6 = "SSI_R6_445NM" 7 = "SSI_R7_753NM" 8 = "SSI_R8_753NM" 9 = "SSI_R9_753NM" 10 = "SSI_R10_604NM" 11 = "SSI_R11_533NM" 12 = "SSI_R12_485NM" SSI (Engineering Model) • Left Eye: 1 = "SSI_L1_445NM" 2 = "SSI_L2_451NM" 3 = "SSI_L3_886NM" 4 = "SSI_L4_OPEN" 5 = "SSI_L5_935NM" 6 = "SSI_L6_672NM" 7 = "SSI_L7_802NM" 8 = "SSI_L8_D_OPEN" 9 = "SSI_L9_901NM" 10 = "SSI_L10_932NM" 11 = "SSI_L11_1001NM" 12 = "SSI_L12_967NM"	**SOURCE** Table lookup **LOCATION** SSI_ARTICULATION_STATE (Group) OM_ARTICULATION_STATE (Group)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		• Right Eye: 1 = "SSI_R1_446NM" 2 = "SSI_R2_671NM" 3 = "SSI_R3_935NM" 4 = "SSI_R4_OPEN" 5 = "SSI_R5_990NM" 6 = "SSI_R6_672NM" 7 = "SSI_R7_753NM" 8 = "SSI_R8_D_OPEN" 9 = "SSI_R9_604NM" 10 = "SSI_R10_533NM" 11 = "SSI_R11_485NM" 12 = "SSI_R12_966NM" <u>MECA-OM Stage Rotation Position</u> Substrate Range: Step Range: "OM1" – "OM3" 0 – 999 "OM4" – "OM6" 1000 – 1999 "OM7" – "OM9" 2000 – 2999 "OM10" – "OM12" 3000 – 3999 "OM13" – "OM15" 4000 – 4999 "OM16" – "OM18" 5000 – 5999 "OM19" – "OM21" 6000 – 6999 "OM22" – "OM24" 7000 – 7999 "OM25" – "OM27" 8000 – 8999 "OM28" – "OM30" 9000 – 9999 "OM31" – "OM33" 10000 – 10999 "OM34" – "OM36" 11000 – 11999 "OM37" – "OM39" 12000 – 12999 "OM40" – "OM42" 13000 – 13999 "OM43" – "OM45" 14000 – 14999 "OM46" – "OM48" 15000 – 15999 "OM49" – "OM51" 16000 – 16999 "OM52" – "OM54" 17000 – 17999 "OM55" – "OM57" 18000 – 18999 "OM58" – "OM60" 19000 – 19999 "OM61" – "OM63" 20000 – 20999 "OM64" – "OM66" 21000 – 21999 "OM67" – "OM69" 22000 – 22999		
ARTICULATION_DEV_POSITION_NAME	Specifies an array of values that provides the formal names for each entry in ARTICULATION_DEV_POSITION. This element names the actual device doing the moving, (e.g., a filter wheel), not the name of a position (e.g., the filter itself).	<u>MECA-OM</u> ("SAMPLE_TYPE", "SUBSTRATE_ID", "SUBSTRATE_POSITION", "FOCAL_DEPTH")	**<u>SOURCE</u>** Static Values **<u>LOCATION</u>** OM_ARTICULATION_STATE (Group)	**<u>TYPE</u>** string array **<u>UNITS</u>** n/a
AZIMUTH_FOV	Specifies the angular measure of the horizontal field of view of an imaged scene.	"0.0" to "360.0"	**<u>SOURCE</u>** Calculation	**<u>TYPE</u>** float

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
			LOCATION INSTRUMENT_STATE_PARMS (Group)	**UNITS** deg (<deg> unit tag required)
BANDS	Specifies the number of spectral bands in image or other object.	Example: "1" = EDR "3" = XYZ RDR or 3-banded multispectral Mosaic RDR	**SOURCE** Constant **LOCATION** IMAGE (Object)	**TYPE** integer **UNITS** n/a
BAND_STORAGE_TYPE	Specifies the storage sequence of lines, samples and bands in an image. The values describe, for example, how different samples are interleaved in image lines, or how samples from different bands are arranged sequentially.	"BAND_SEQUENTIAL", "SAMPLE_INTERLEAVED", "LINE_INTERLEAVED"	**SOURCE** Constant **LOCATION** IMAGE (Object)	**TYPE** string(20) **UNITS** n/a
BYTES	Specifies the number of bytes allocated for a particular data representation.	"0" to n	**SOURCE** Calculation: • Based on size of VICAR label. **LOCATION** IMAGE_HEADER (Object)	**TYPE** integer **UNITS** n/a
CALIBRATION_SOURCE_ID	Specifies a unique identifier (within a data set) indicating the source of the calibration data used in generating the entity described by the enclosing group (often, a camera model). The construction of this identifier is mission-specific, but should indicate which specific calibration data set was used (via date or other means) and may also indicate the calibration method.		**SOURCE** Ground-derived conversion to string. **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** string(47) **UNITS** n/a
CHECKSUM	Specifies an unsigned 32-bit sum of all data values in a data object.	"0" to "4294967295"	**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** integer **UNITS** n/a
AC_CMPRS_TABLE_ID CMPRS_AC_INDEX [V]	The index used to select the JPEG AC-Huffman Luminance Bits lookup table for compression.	"0" to "2"	**SOURCE** comp_ac_idx **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer **UNITS** n/a
DC_CMPRS_TABLE_ID CMPRS_DC_INDEX [V]	The index used to select the JPEG DC-Huffman Luminance Bits lookup table for compression.	"0" to "2"	**SOURCE** comp_dc_idx **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
CMPRS_Q_INDEX	The table index used to select the JPEG quantization lookup table for compression.	“0”, “1”	**SOURCE** comp_q_idx **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer **UNITS** n/a
CONFIGURATION_BAND_ID	Specifies an array of strings identifying the configuration of the IDD arm represented by the corresponding band in the image. The first entry in the array identifies the configuration for the first band, the second entry for the second band, etc. See also INSTRUMENT_BAND_ID.	Scoop 1 = "ELBOW_UP” 2 = "ELBOW_DOWN” Scoop_BTM 3 = "ELBOW_UP” 4 = "ELBOW_DOWN” Blade 5 = "ELBOW_UP” 6 = "ELBOW_DOWN” ISAD1 7 = "ELBOW_UP” 8 = "ELBOW_DOWN” ISAD2 9 = "ELBOW_UP” 10 = "ELBOW_DOWN” TECP 11 = "ELBOW_UP” 12 = "ELBOW_DOWN”	**SOURCE** Static Values **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array[16] **UNITS** n/a
COORDINATE_SYSTEM_INDEX	Specifies an integer array used to record and track the movement of a rover or lander during surface operations. When in a COORDINATE_SYSTEM_STATE group, this keyword identifies which instance of the coordinate frame, named by COORDINATE_SYSTEM_NAME, is being defined by the group. For PHX, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the lander may potentially have moved, e.g. due to arm motion. The full counter may have up to 2 values (SITE, POSITION), but normally only the first value (for SITE frames) or the first two values (for LOCAL_LEVEL or PAYLOAD frames) are used for defining coordinate system instances. It is legal to use any number of indices to describe a coordinate system		**SOURCE** Group Dependent, Table Lookup Calculation: **LOCATION** Group Dependent: a) PAYLOAD_COORDINATE_SYSTEM (Group) b) LOCAL_LEVEL_COORDINATE_SYSTEM (Group) c) SITE_COORDINATE_SYSTEM (Group)	**TYPE** integer array[2] **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	instance, however. Example: COORDINATE_SYSTEM_INDEX=(1,1)			
COORDINATE_SYSTEM_INDEX_NAME	Specifies an array of the formal names identifying each integer specified in COORDINATE_SYSTEM_INDEX.	"SITE", "POSITION"	**SOURCE** Group Dependent: • Static Value **LOCATION** Group Dependent: a) PAYLOAD_COORDINATE_SYSTEM (Group) b) LOCAL_LEVEL_COORDINATE_SYSTEM (Group) c) SITE_COORDINATE_SYSTEM (Group)	**TYPE** string array[2] **UNITS** n/a
COORDINATE_SYSTEM_NAME	Specifies the full name of the coordinate system to which the state vectors are referenced. When in a COORDINATE_SYSTEM group, this keyword provides the full name of the coordinate system being defined by the group. The rest of the keywords in the group describe how this coordinate system is related to some other (the "reference"). Non-unique coordinate systems (such as "SITE" for rover or lander missions), which have multiple instances using the same name, also require COORDINATE_SYSTEM_INDEX to completely identify the coordinate system.	"SITE_FRAME", "PAYLOAD_FRAME", "LOCAL_LEVEL_FRAME"	**SOURCE** Group Dependent: • Static Value **LOCATION** Group Dependent: a) PAYLOAD_COORDINATE_SYSTEM (Group) b) LOCAL_LEVEL_COORDINATE_SYSTEM (Group) c) SITE_COORDINATE_SYSTEM (Group)	**TYPE** string(30) **UNITS** n/a
DARK_CURRENT_FILE_NAME	Specifies a list of the names of the dark current files used in generating the RDR.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array **UNITS** n/a
DARK_CURRENT_FILE_DESC DARK_CURRENT_FILE_NAME_DESC [V]	Specifies a description of the corresponding dark current files listed in DARK_CURRENT_FILE.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string **UNITS** n/a
DATA_SET_ID	Specifies a unique alphanumeric identifier for a data set or a data product. The DATA_SET_ID value for a given data set or product is constructed according to flight project naming conventions. In most cases the DATA_SET_ID is an abbreviation	<u>EDRs</u> "PHX-M-SSI-2-EDR-V1.0", "PHX-M-RAC-2-EDR-V1.0", "PHX-M-OM-2-EDR-V1.0" <u>"Operations" RDRs (SSI)</u> "PHX-M-SSI-3-RADIOMETRIC-OPS-	**SOURCE** • PDS • Table Lookup **LOCATION** IDENTIFICATION (Class)	**TYPE** string(40) **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	of the DATA_SET_NAME. In the PDS, the values for DATA_SET_ID are constructed according to standards outlined in the Standards Reference.	V1.0", "PHX-M-SSI-4-LINEARIZED-OPS-V1.0", "PHX-M-SSI-5-DISPARITY-OPS-V1.0", "PHX-M-SSI-5-XYZ-OPS-V1.0", "PHX-M-SSI-5-NORMAL-OPS-V1.0", "PHX-M-SSI-5-RANGE-OPS-V1.0", "PHX-M-SSI-5-ROUGHNESS-OPS-V1.0", "PHX-M-SSI-5-REACHABILITY-OPS-V1.0", "PHX-M-SSI-5-MESH-OPS-V1.0", "PHX-M-SSI-5-MOSAIC-OPS-V1.0", "PHX-M-SSI-5-ANAGLYPH-OPS-V1.0" "Operations" RDRs (RAC) "PHX-M-RAC-3-RADIOMETRIC-OPS-V1.0", "PHX-M-RAC-4-LINEARIZED-OPS-V1.0", "PHX-M-RAC-5-DISPARITY-OPS-V1.0", "PHX-M-RAC-5-XYZ-OPS-V1.0", "PHX-M-RAC-5-NORMAL-OPS-V1.0", "PHX-M-RAC-5-RANGE-OPS-V1.0", "PHX-M-RAC-5-ROUGHNESS-OPS-V1.0", "PHX-M-RAC-5-REACHABILITY-OPS-V1.0", "PHX-M-RAC-5-MESH-OPS-V1.0", "PHX-M-RAC-5-MOSAIC-OPS-V1.0", "PHX-M-RAC-5-ANAGLYPH-OPS-V1.0" "Operations" RDRs (other) "PHX-M-RVRCAM-6-RMC-OPS-V1.0" "Science" RDRs (SSI) "PHX-M-SSI-3-RADIOMETRIC-SCI-V1.0", "PHX-M-SSI-5-MOSAIC-SCI-V1.0", "PHX-M-SSI-5-ANAGLYPH-SCI-V1.0" "PHX-M-SSI-5-IOF-SCI-V1.0"		

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		“Science” RDRs (RAC) “PHX-M-RAC-3-RADIOMETRIC-SCI-V1.0”, “PHX-M-RAC-5-MOSAIC-SCI-V1.0”, “PHX-M-RAC-5-ANAGLYPH-SCI-V1.0” “Science” RDRs (MECA-OM) “PHX-M-OM-3-RADIOMETRIC-SCI-V1.0		
DATA_SET_NAME	Specifies the full name given to a data set or a data product. The DATA_SET_NAME typically identifies the instrument that acquired the data, the target of that instrument, and the processing level of the data. In the PDS, values for DATA_SET_NAME are constructed according to standards outlined in the Standards Reference.	EDRs "PHOENIX MARS SURFACE STEREO IMAGER 2 EDR VERSION 1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 2 EDR VERSION 1.0", "PHOENIX MARS MECA OPTICAL MICROSCOPE 2 EDR VERSION 1.0" “Operations” RDRs (SSI) "PHOENIX MARS SURFACE STEREO IMAGER 3 RADIOMETRIC OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 4 LINEARIZED OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 DISPARITY OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 XYZ OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 NORMAL OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 RANGE OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 ROUGHNESS OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 REACHABILITY OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 TERRAIN MESH OPS V1.0",	**SOURCE** • PDS • Table Lookup **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"PHOENIX MARS SURFACE STEREO IMAGER 5 MOSAIC OPS V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 ANAGLYPH OPS V1.0" “Operations” RDRs (RAC) "PHOENIX MARS ROBOTIC ARM CAMERA 3 RADIOMETRIC OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 4 LINEARIZED OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 DISPARITY OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 XYZ OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 NORMAL OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 RANGE OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 ROUGHNESS OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 REACHABILITY OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 TERRAIN MESH OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 MOSAIC OPS V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 ANAGLYPH OPS V1.0" “Operations” RDRs (other) "PHOENIX MARS RVRCAM 6 ROVER MOTION COUNTER OPS V1.0" “Science” RDRs (SSI) "PHOENIX MARS SURFACE STEREO IMAGER 3 RADIOMETRIC SCI V1.0", "PHOENIX MARS SURFACE STEREO IMAGER 5 MOSAIC SCI V1.0", "PHOENIX MARS SURFACE		

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		STEREO IMAGER 5 ANAGLYPH SCI V1.0" "PHOENIX MARS SURFACE STEREO IMAGER 5 INTEN OVER FLX SCI V1.0" “Science” RDRs (RAC) "PHOENIX MARS ROBOTIC ARM CAMERA 3 RADIOMETRIC SCI V1.0", “PHOENIX MARS ROBOTIC ARM CAMERA 5 MOSAIC SCI V1.0", "PHOENIX MARS ROBOTIC ARM CAMERA 5 ANAGLYPH SCI V1.0" “Science” RDRs (MECA-OM) "PHOENIX MARS MECA OPTICAL MICROSCOPE RADIOM RDR SCI V1.0"		
DERIVED_IMAGE_TYPE	Specifies an interpret the pixel values in a derived image RDR (or colloquially, the type of the derived image itself). Values are defined as: IMAGE – Standard image, where pixels represent intensity. Note: This implies nothing about radiometric, geometric, or other corrections that may have been applied. DISPARITY_MAP – Pixels represent line and sample disparity with respect to another image (2 bands). DISPARITY_LINE_MAP – Pixels represent line disparity only. DISPARITY_SAMPLE_MAP – Pixel represent sample disparity only. MASK_MAP – Pixels represent a mask, indicating valid/invalid regions according to mask criteria. XYZ_MAP – Pixels represent XYZ values (3 bands). X_MAP – Pixels represent the X	“IMAGE”, “DISPARITY_MAP”, “DISPARITY_LINE_MAP”, “DISPARITY_SAMPLE_MAP”, “XYZ_MAP”, “X_MAP”, “Y_MAP”, “Z_MAP”, “RANGE_MAP”, “UVW_MAP”, “U_MAP”, “V_MAP”, “W_MAP”, “REACHABILITY_MAP”, “MASK_MAP”	**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	component of an XYZ image. Y_MAP – Pixels represent the Y component of an XYZ image. Z_MAP – Pixels represent the Z component of an XYZ image. RANGE_MAP – Pixels represent a distance from the camera center. UVW_MAP – Pixels represent Surface Normal values (3 bands associating to X, Y, Z). U_MAP – Pixels represent the U (X) component of a Surface Normal image. V_MAP – Pixels represent the V (Y) component of a Surface Normal image. W_MAP – Pixels represent the W (Z) component of a Surface Normal image. REACHABILITY_MAP – Pixels flag what is reachable on the target feature by the respective IDD instrument.			
^DESCRIPTION	Specifies a pointer that provides a free-form, unlimited-length character string that represents or gives an account of something.	"VICAR2.TXT"	**SOURCE** Static Value **LOCATION** IMAGE_HEADER (Object)	**TYPE** string **UNITS** n/a
DETECTOR_FIRST_LINE	Specifies the starting row from the hardware, such as a charge-coupled device (CCD), that contains data. For PHX, the Y position of the start of the CCD window in pixels for SSI only.	"1" to "1024"	**SOURCE** ccd_window_end **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a
DETECTOR_LINES	Specifies the number of rows extracted from the hardware, such as a charge-coupled device (CCD), that contain data. For PHX, records the Y (row) end position of a sub-framed image with a default of "0" for images that are not sub-framed.	"0" to "1024"	**SOURCE** • Sub-framed images: subframe_y_end • Images that are not sub-framed: image_height **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a
DOWNSAMPLE_METHOD	Specifies whether or not hardware downsampling was applied to an image.	<u>SSI</u> "NONE", "HARDWARE",	**SOURCE** Calculation:	**TYPE** string

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	For PHX, depends on combination of 2 values: a) T/F = Hardware binning on CCD for SSI only. b) Amount and type of SW decimation applied.	"SOFTWARE", "BOTH" RAC and OM "NONE", "SOFTWARE"	• ccd_binning_mode • image_dec_mode NOTES: • SSI "NONE" - ccd_binning_mode=0 and image_dec_mode=0 "HARDWARE" - ccd_binning_mode=1 and image_dec_mode=0 "SOFTWARE" - ccd_binning_mode=0 and image_dec_mode=2,3,4 "BOTH" - ccd_binning_mode=1 and image_dec_mode=1 • RAC and OM "NONE" - image_dec_mode=0 "SOFTWARE" - image_dec_mode does not equal "0" **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**UNITS** n/a
EARTH_RECEIVED_START_TIME	Specifies the beginning time at which telemetry was received during a time period of interest. This should be represented in UTC system format.	YYYY-MM-DDThh:mm:ss[.fff]Z	**SOURCE** Calculation: • From CCSDS packet header **LOCATION** TELEMETRY (Class)	**TYPE** datetime **UNITS** n/a
EARTH_RECEIVED_STOP_TIME	Specifies the ending time for receiving telemetry during a time period of interest. This should be represented in UTC system format.	YYYY-MM-DDThh:mm:ss[.fff]Z	**SOURCE** Calculation: • From CCSDS packet header **LOCATION** TELEMETRY (Class)	**TYPE** datetime **UNITS** n/a
ELEVATION_FOV	Specifies the angular measure of the horizontal field of view of an imaged scene.	"0.0" to "360.0"	**SOURCE** Calculation **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
ERROR_PIXELS	Specifies the number of pixels that are outside a valid DN range, after all decompression and post decompression processing has been completed.		**SOURCE** Calculated by telemetry **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer **UNITS** n/a
EXPECTED_PACKETS	Specifies the total number of telemetry packets which constitute a complete data product, i.e., a data product without missing		**SOURCE** packet_count	**TYPE** integer

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	data.		**LOCATION** TELEMETRY (Class)	**UNITS** n/a
EXPOSURE_COUNT	Specifies the maximum number of exposures taken during a specified interval. The value is dependent on exposure type. For PHX, specifies the number of auto-exposure attempts made before either (a) a suitable exposure was identified or (b) the auto-exposure function reached the attempt limit specified in its configuration file. A value of zero in this field indicates that auto-exposure was not used for this image.	"0" to "65535"	**SOURCE** autoexpose_count **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a
EXPOSURE_DURATION	Specifies the value of the time between the opening and closing of an instrument aperture (such as a camera shutter). For PHX, it is image exposure time in milliseconds, which is calculated depending on the instrument as follows: For SSI instrument: Exp Time = Number of camera time intervals * 5.12 ms For RAC and OM instruments: Exp Time = Number of camera time intervals * 0.5 ms		**SOURCE** Calculation based on instrument: • SSI exp_time_counts * 5.12 ms • RAC and OM exp_time_counts * 0.5 ms **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** float **UNITS** ms (<ms> unit tag required)
EXPOSURE_DURATION_COUNT	Specifies the value, in raw counts, of the time interval between the opening and closing of an instrument aperture (such as a camera shutter). This is a raw value taken directly from telemetry, as opposed to EXPOSURE_DURATION, which has been converted to engineering units. For PHX, it is image exposure time in camera time intervals. For SSI instrument: 1 Camera Time Interval = 5.12 ms For RAC and OM instruments: 1 Camera Time Interval = 0.5 ms	"0" to "65535"	**SOURCE** exp_time_counts **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a
FILE_RECORDS	Specifies the number of physical file records, including both label records and data records. Note: In the PDS the use of	"0" to n	**SOURCE** Calculation **LOCATION**	**TYPE** integer **UNITS**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	FILE_RECORDS along with other file-related data elements is fully described in the Standards Reference.		FILE DATA ELEMENT (Class)	n/a
FILTER_NAME	Specifies the commonly-used name of the instrument filter through which an image or measurement was acquired or which is associated with a given instrument mode. See also FILTER_NUMBER. NOTE: FILTER_NAME is unique, while the FILTER_NUMBER is not.	Flight Model (FM) • SSI Left Eye: "SSI_L1_672NM" "SSI_L2_445NM" "SSI_L3_451NM" "SSI_L4_990NM" "SSI_L5_886NM" "SSI_L6_830NM" "SSI_L7_802NM" "SSI_L8_861NM" "SSI_L9_901NM" "SSI_L10_932NM" "SSI_L11_1001NM" "SSI_L12_967NM" • SSI Right Eye: "SSI_R1_672NM" "SSI_R2_445NM" "SSI_R3_671NM" "SSI_R4_936NM" "SSI_R5_936NM" "SSI_R6_445NM" "SSI_R7_753NM" "SSI_R8_753NM" "SSI_R9_753NM" "SSI_R10_604NM" "SSI_R11_533NM" "SSI_R12_485NM" Engineering Model (EM) • SSI Left Eye: "SSI_L1_445NM" "SSI_L2_451NM" "SSI_L3_886NM" "SSI_L4_OPEN" "SSI_L5_935NM" "SSI_L6_672NM" "SSI_L7_802NM" "SSI_L8_D_OPEN" "SSI_L9_901NM" "SSI_L10_932NM" "SSI_L11_1001NM" "SSI_L12_967NM" • SSI Right Eye: "SSI_R1_446NM"	**SOURCE** Group Dependent, Table Lookup **LOCATION** Group Dependent: a) GEOMETRIC_CAMERA_MODEL (Group) b) INSTRUMENT_STATE_PARMS (Group)	**TYPE** string array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"SSI_R2_671NM" "SSI_R3_935NM" "SSI_R4_OPEN" "SSI_R5_990NM" "SSI_R6_672NM" "SSI_R7_753NM" "SSI_R8_D_OPEN" "SSI_R9_604NM" "SSI_R10_533NM" "SSI_R11_485NM" "SSI_R12_966NM"		
FILTER_NUMBER	Specifies the number of an instrument filter through which an image or measurement was acquired or which is associated with a given instrument mode. For PHX, current filter wheel filter ID number for SSI only. See also FILTER_NAME. Note: FILTER_NAME is unique, while the FILTER_NUMBER is not.	SSI Left Eye "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12" SSI Right Eye "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"	**SOURCE** fw_pos_filt_curr **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer array **UNITS** n/a
FIRST_LINE	Specifies the line within a source image that corresponds to the first line in a sub-image.	SSI "1" to "1024" RAC & MECA-OM "1" to "512"	**SOURCE** subframe_y_start + 1 **LOCATION** Group Dependent: a) SUBFRAME_PARMS (Group) b) IMAGE (Object)	**TYPE** integer **UNITS** n/a
FIRST_LINE_SAMPLE	Specifies the sample within a source image that corresponds to the first sample in a sub-image.	SSI "1" to "1024" RAC & MECA-OM "1" to "256"	**SOURCE** subframe_x_start + 1 **LOCATION** Group Dependent: a) SUBFRAME_PARMS (Group) b) IMAGE (Object)	**TYPE** integer **UNITS** n/a
FLAT_FIELD_FILE_DESC FLAT_FIELD_FILE_NAME [V]	Specifies the array of names of the flat-field files used in generating the RDR.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array **UNITS** n/a
FLAT_FIELD_FILE_NAME_DESC	Specifies a description of the corresponding flat field files listed in FLAT_FIELD_FILE.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
FRAME_ID	Specifies an identification for a particular instrument measurement frame. A frame consists of a sequence of measurements made over a specified time interval, and may include measurements from different instrument modes. These sequences repeat from cycle to cycle and sometimes within a cycle. Note that mosaics may contain more than one value in an array.	"LEFT", "RIGHT", "MONO"	**SOURCE** Based on source ID. **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a
FRAME_TYPE	Specifies whether the image was commanded as part of a stereo pair or as a single left or right monoscopic image. If FRAME_TYPE=STEREO, a left and a right image should be present for the same IMAGE_ID	0 = "MONO" 1 = "STEREO"	**SOURCE** source_id **LOCATION** IDENTIFICATION (Class)	**TYPE** string(10) **UNITS** n/a
GEOMETRY_PROJECTION_TYPE	Specifies the state of the pixels in an image before a re-projection has been applied. Describes if or how the pixels have been reprojected. RAW indicates reprojection has not been done; the pixels are as they came from the camera. This means the image uses a CAHVOR or one of the CAHVORE camera models. LINEARIZED means that reprojection has been performed to linearize the camera model (thus removing things like lens distortion). For PHX, in the Linearized case, the SSI and RAC instrument images use a CAHV camera model.	"RAW", "LINEARIZED"	**SOURCE** Image Processing Software **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a
HEADER_TYPE	Specifies a specific type of header data structure. For example: FITS, VICAR. Note: In the PDS, HEADER_TYPE is used to indicate non-PDS headers.	"VICAR2"	**SOURCE** Static Value **LOCATION** IMAGE_HEADER (Object)	**TYPE** string(12) **UNITS** n/a
^IMAGE	Specifies a pointer to the IMAGE object. See chapter 14 of the PDS Standards Reference for more information on pointer usage.		**SOURCE** Calculation **LOCATION** POINTERS	**TYPE** NULL **UNITS** n/a
ELECTRONICS_BIAS IMAGE_BIAS [V]	Bias value added to images during onboard corrections.		**SOURCE** image_bias	**TYPE** integer

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
			LOCATION INSTRUMENT_STATE_PARMS (Group)	**UNITS** n/a
^IMAGE_HEADER	Specifies a pointer to the IMAGE_HEADER object. See chapter 14 of the PDS Standards Reference for more information on pointer usage.		**SOURCE** Calculation **LOCATION** POINTERS	**TYPE** NULL **UNITS** n/a
IMAGE_ID	Specifies an image and typically consists of a sequence of characters representing 1) a routinely occurring measure, such as revolution number, 2) a letter identifying the spacecraft, target, or camera, and 3) a representation of a count within the measure, such as picture number within a given revolution. For PHX, this is to be the 32-bit “token” value. The PHX “token” is comprised of 3 components: a) Campaign, b) Activity, and c) Image ID. Example: Mariner 9 - LevanthalIdentifier - (orbit, camera, pic #, total # of pics in orbit) Viking Orbiter - (orbit #, sc, pic # (FSC/16)), Viking Lander - (sc, camera, mars doy, diode (filter), pic # for that day), Voyager - (pic # for encounter, FDS for cruise)		**SOURCE** img_id_number **LOCATION** IDENTIFICATION (Class)	**TYPE** string(30) **UNITS** n/a
IMAGE_TYPE	Specifies the type of image acquired. This may be used to describe characteristics that differentiate one group of images from another such as the nature of the data in the image file, the purpose for which the image was acquired, or the way in which it was acquired. This element is very similar to the older image_observation_type element, but is designed to resolve ambiguities in cases where missions utilize a naming convention for both specific images and more general observations, which consist of multiple images. In those cases, the latter may be described by the observation_type element.	“REGULAR”	**SOURCE** Static **LOCATION** IDENTIFICATION (Class)	**TYPE** string(15) **UNITS** n/a
SOURCE_PRODUCT_ID INPUT_IMAGE_SET	Specifies a list of the PRODUCT_IDs of images used to generate this RDR.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
INSTRUMENT_AZIMUTH	Specifies the value for an instrument's rotation in the horizontal direction. It is usually measured from some kind of low hard stop. Although it may be used for any instrument where it makes sense, it is primarily intended for use in surface-based instruments that measure pointing in terms of azimuth and elevation. When in a DERIVED_GEOMETRY group, defines the azimuth (horizontal rotation) at which the instrument is pointed. This value is expressed using the cooridinate system referred to by REFERENCE_COORD_SYSTEM_NAME and REFERENCE_COORD_SYSTEM_INDEX contained within the same group. The interpretation of exactly what part of the instrument is being pointed is mission-specific. It could be the boresight, the camera head direction, the CAHV camera model A vector direction, or any of a number of other things. As such, for multimission use this value should be used mostly as an approximation, e.g. identifying scenes which might contain a given object. For Phoenix, the azimuth is defined as the result of projecting the center pixel of the image through the camera model. The resulting vector's azimuth is used. Note that the center of the image Is (512,512) for an SSI Full Frame, but may be different fo Subframed or Downsampled images, and for other cameras.		**SOURCE** Calculation **LOCATION** • PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) • SITE_DERIVED_GEOMETRY_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
INSTRUMENT_BAND_ID	Specifies an array of strings identifying the instrument or tool frame represented by the corresponding band in the image. The first entry in the array identifies the instrument for the first band, the second entry for the second band, etc. See also CONFIGURATION_BAND_ID.	BAND VALUE 1 "SCOOP" 2 "SCOOP_BTM" 3 "BLADE" 4 "ISAD1" 5 "ISAD2" 6 "RAC" 7 "TECP" 8 "WRIST"	**SOURCE** Static Value **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array[16] **UNITS** n/a
INSTRUMENT_ELEVATION	Specifies a value for an instrument's rotation in the vertical direction. It is usually measured from some kind of low hard stop.		**SOURCE** Calculation	**TYPE** float

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Although it may be used for any instrument where it makes sense, it is primarily intended for use in surface-based instruments that measure pointing in terms of azimuth and elevation. When in a DERIVED_GEOMETRY group, defines the elevation (vertical rotation) at which the instrument is pointed. This value is expressed using the cooridinate system referred to by REFERENCE_COORD_SYSTEM_NAME and REFERENCE_COORD_SYSTEM_INDEX contained within the same group. The interpretation of exactly what part of the instrument is being pointed is mission-specific. It could be the boresight, the camera head direction, the CAHV camera model A vector direction, or any of a number of other things. As such, for multimission use this value should be used mostly as an approximation, e.g. identifying scenes which might contain a given object. For Phoenix, the elevation is defined as the result of projecting the center pixel of the image through the camera model. The resulting vector's elevation is used. Note that the center of the image Is (512,512) for an SSI Full Frame, but may be different fo Subframed or Downsampled images, and for other cameras.		**LOCATION** • PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) • SITE_DERIVED_GEOMETRY_PARMS (Group)	**UNITS** deg (<deg> unit tag required)
INSTRUMENT_HOST_ID	Specifies a unique identifier for the host where an instrument is located.	VALUE "PHX"	**SOURCE** Static value **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a
INSTRUMENT_HOST_NAME	Specifies the full name of the host on which an instrument is based. This host can be either a spacecraft or an earth base. Thus, the INSTRUMENT_HOST_NAME element can contain values which are either SPACECRAFT_NAME values or EARTH_BASE_NAME values. Note that mosaics may contain more than one value in an array.	"PHOENIX LANDER"	**SOURCE** Static value **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
INSTRUMENT_ID	Specifies an abbreviated name or acronym which identifies an instrument. Note: INSTRUMENT_ID is not a unique identifier for a given instrument. Note also that the associated INSTRUMENT_NAME element provides the full name of the instrument. Example values: IRTM (for Viking Infrared Thermal Mapper), PWS (for plasma wave spectrometer).	SOURCE ID VALUE 0 "SSI (per " 1 "SSI" 2 "RAC" 3 "OM"	**SOURCE** • source_id • data_Type **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a
INSTRUMENT_MODE_ID	Specifies an instrument-dependent designation of operating mode. This may be simply a number, letter or code, or a word such as 'normal', 'full resolution', 'near encounter', or 'fixed grating'.	"FULL_FRAME", "WINDOWED_FRAME", "4X1SUMMATION_FRAME", "2X2DECIMATION_MODE", "3X3DECIMATION_MODE", "4X4DECIMATION_MODE"	**SOURCE** Calculation: • source_id • dec_mode_avg • subframe_x_end • subframe_y_end NOTES: • If dec_mode_avg = 0 - and source_id = SSI and (subframe_y_end < 1023 or subframe_x_end < 1023), then value is "WINDOWED_FRAME" - and (source_id = RAC or source_id = OM) and (subframe_y_end < 255 or subframe_x_end < 511), then value is "WINDOWED_FRAME" - otherwise, value is "FULL_FRAME" • If dec_mode_avg = 1, then value is "4X1SUMMATION_FRAME" • If dec_mode_avg = 2, then value is "2X2DECIMATION_MODE" • If dec_mode_avg = 3, then value is "3X3DECIMATION_MODE" • If dec_mode_avg = 4, then value is "4X4DECIMATION_MODE" **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string(20) **UNITS** n/a
INSTRUMENT_NAME	Specifies the full name of an instrument. Note: that the associated INSTRUMENT_ID element provides an abbreviated name or acronym for the instrument.	"SURFACE STEREO IMAGER RIGHT", "Robotic Arm Camera", "MECA Optical Microscope"	**SOURCE** source_id **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Example values: FLUXGATE MAGNETOMETER, NEAR_INFRARED MAPPING SPECTROMETER.			
INSTRUMENT_SERIAL_NUMBER	Specifies the manufacturer's serial number assigned to an instrument. This number may be used to uniquely identify a particular instrument for tracing its components or determining its calibration history, for example.	"1" to "255"	**SOURCE** Table Lookup: • Based on INSTRUMENT_HOST_ID and SOURCE_ID. **LOCATION** IDENTIFICATION (Class)	**TYPE** integer **UNITS** n/a
INSTRUMENT_TEMPERATURE	Specifies the temperature, in degrees Celsius (deg C), of an instrument or some part of an instrument. Note that this may be an array of multiple values for temperatures on different parts of the instrument. (Example: CCD array and sensor head). For PHX, temperatures in deg C are derived from raw readout counts applied in equations. For SSI instrument, array elements: 1) Temperature in deg C at start of exposure. 2) Temperature in deg C at end of exposure. 3) Temperature in deg C of optical bench. 4) Temperature in deg C of AZ actuator at start of exposure. 5) Temperature in deg C of EL actuator at start of exposure. 6) Temperature in deg C of FW actuator at start of exposure. 7) Temperature in deg C of PCB For RAC instrument, array elements: 1) Temperature in deg C of focus motor at start of exposure. 2) Temperature in deg C of optical bench. For MECA-OM instrument EDRs, array elements: 1) Temperature in deg C at start of exposure. 2) Temperature in deg C at end of exposure. See also INSTRUMENT_TEMPERATURE_COUNT		**SOURCE** Calculation based on instrument: • For SSI Left CCD, array elements: 1) (-3.20817e-07 * (ccd_temp_start_raw * ccd_temp_start_raw)) + (5.02592e-02 * ccd_temp_start_raw) + (-24.941) 2) (-3.20817e-07 * (ccd_temp_end_raw * ccd_temp_end_raw)) + (5.02592e-02 * ccd_temp_end_raw) + (-24.941) 3) (-2.75874e-06 * (optical_bench_temp_raw * optical_bench_temp_raw)) + (8.82068e-02 * optical_bench_temp_raw) + (-104.07) 4) (1.52191e-01 * az_temp_raw) + 7.76619e-01 5) (1.54829e-01 * el_temp_raw) + 7.00375e-01 6) ((2048 * 18.307) / (fw_temp_raw + 2048) - 76.368) / (76.368 * 3.8388e-03) + 29.304 7) (-3.77049e-08 * (ccd_temp_pcb_raw * ccd_temp_pcb_raw)) + (4.97748e-02 * ccd_temp_pcb_raw) + (-23.911) • For SSI Right CCD, array elements: 1) (-1.87170e-07 * (ccd_temp_start_raw * ccd_temp_start_raw)) + (5.00252e-02 * ccd_temp_start_raw) + (-24.593) 2) (-1.87170e-07 * (ccd_temp_end_raw * ccd_temp_end_raw)) + (5.00252e-02 * ccd_temp_end_raw) + (-24.593) 3) (-2.75874e-06 * (optical_bench_temp_raw * optical_bench_temp_raw)) + (8.82068e-02 * optical_bench_temp_raw) + (-104.07) 4) (1.52191e-01 * az_temp_raw) + 7.76619e-01 5) (1.54829e-01 * el_temp_raw) + 7.00375e-01 6) ((2048 * 18.307) / (fw_temp_raw + 2048) -	**TYPE** float array **UNITS** deg C (<degC> unit tag required)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	and INSTRUMENT_TEMPERATURE_NAME.		76.368) / (76.368 * 3.8388e-03) + 29.304 7) (2.93470e-07 * (ccd_temp_pcb_raw * ccd_temp_pcb_raw)) + (5.04260e-02 * ccd_temp_pcb_raw) + (-24.679) • For RAC, array elements: 1) (foc_temp_raw * 0.11883) + (-268.555) 2) (optical_bench_temp_raw * 0.08293) + (-273.643) • For MECA-OM, array elements: 1) ((ccd_temp_start_raw + 2048) / 4096) * 200 - 125 2) ((ccd_temp_end_raw + 2048) / 4096) * 200 - 125 **LOCATION** INSTRUMENT_STATE_PARAMS (Group)	
INSTRUMENT_TEMPERATURE_COUNT	Specifies the temperature, in raw readout counts, of an instrument or some part of an instrument. Note that this may be an array of multiple values for temperatures on different parts of the instrument. (Example: CCD array and sensor head). For SSI instrument, array elements: 1) Raw temp readout at start of exposure. 2) Raw temp readout at end of exposure. 3) Raw temp readout of optical bench. 4) Raw temp of AZ actuator at start of exposure. 5) Raw temp of EL actuator at start of exposure. 6) Raw temp of FW actuator at start of exposure. 7) Raw temp readout of PCB. For RAC instrument, array elements: 1) Raw temp of focus motor at start of exposure. 2) Raw temp readout of optical bench. For MECA-OM instrument, array elements: 1) Raw temp readout at start of exposure. 2) Raw temp readout at end of exposure. See also INSTRUMENT_TEMPERATURE and INSTRUMENT_TEMPERATURE_NAME.	-2048 .. +2047	**SOURCE** Instrument Dependent: • For SSI, array elements: 1) ccd_temp_start_raw 2) ccd_temp_end_raw 3) optical_bench_temp_raw 4) az_temp_raw 5) el_temp_raw 6) fw_temp_raw 7) ccd_temp_pcb_raw • For RAC, array elements: 1) foc_temp_raw 2) optical_bench_temp_raw • For MECA-OM, array elements: 1) ccd_temp_start_raw 2) ccd_temp_end_raw **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
INSTRUMENT_TEMPERATURE_NAME	Specifies an array of the formal names identifying each of the values used in INSTRUMENT_TEMPERATURE and INSTRUMENT_TEMPERATURE_COUNT. See also INSTRUMENT_TEMPERATURE and INSTRUMENT_TEMPERATURE_COUNT.	(“SSI_CCD_START”, “SSI_CCD_END”, “SSI_OPTICAL_BENCH”, “SSI_AZIMUTH_ACTUATOR”, “SSI_ELEVATION_ACTUATOR”, “SSI_FILTER_WHEEL_ACTUATOR” ,”SSI_PBC_END”, “RAC_FOCUS_MOTOR”, “RAC_OPTICAL_BENCH”, “OM_CCD_START”, “OM_CCD_END”)	**SOURCE** Static Value **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string array[9] **UNITS** n/a
INSTRUMENT_TYPE	Specifies the type of an instrument. Example values: POLARIMETER, RADIOMETER, REFLECTANCE SPECTROMETER, VIDICON CAMERA. Note that mosaics may contain more than one value in an array.	“IMAGING CAMERA”, “ATOMIC FORCE MICROSCOPE”	**SOURCE** Static Value **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string array **UNITS** n/a
INSTRUMENT_VERSION_ID	Specifies the model of an instrument used to obtain data. For example, this keyword could be used to distinguish between an engineering model of a camera used to acquire test data, and a flight model of a camera used to acquire science data during a mission.	SCID Phase Value 84 Cruise “FM” 84 Ops “FM” 89 Test “EM”	**SOURCE** Table Lookup **LOCATION** IDENTIFICATION (Class)	**TYPE** string(8) **UNITS** n/a
INST_CMPRS_MODE	Specifies the method used for on-board compression of data. Note: The INST_CMPRS_NAME element provides the full name of an INST_CMPRS_MODE.	“0” = None “1” = JPEG Huffman encoding “2” = JPEG Arithmetic encoding “3” = RICE compression “4” = LUT3 reduction “5” = SX4 reduction	**SOURCE** • comp_done • comp_mode NOTE: Definition of the 2 sources: • Flag to indicate if image data is actually compressed or not (T/F). (COMP_DONE). • Code indicating the compression mode requested for the image (COMP_MODE). **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** string **UNITS** n/a
INST_CMPRS_NAME	Specifies the type of on-board compression used for data storage and transmission. Note: The INST_CMPRS_MODE element provides an abbreviated identifier for the INST_CMPRS_NAME.	“None”, “JPEG with Huffman Encoding”, “JPEG with Arithmetic Encoding, “Rice compression”, “LUT3 reduction”, “SX4 (shift-by-four) reduction”	**SOURCE** comp_mode **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** string **UNITS** n/a
INST_CMPRS_QUALITY	Specifies a JPEG- or ICER-specific variable which identifies the resultant or targeted	“0” to “65535”	**SOURCE** comp_q	**TYPE** integer

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	image quality index for on-board data compression. For PHX, Q value used for JPEG only.		**LOCATION** COMPRESSION_PARMS (Group)	**UNITS** n/a
INST_CMPRS_RATE	Specifies the average number of bits needed to represent a pixel for an on-board compressed image.	"0" to "12"	**SOURCE** Calculation **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** float **UNITS** n/a
INST_CMPRS_RATIO	Specifies the ratio of the size, in bytes, of the original uncompressed data file to its compressed form.		**SOURCE** comp_level **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** float **UNITS** n/a
INST_CMPRS_SEGMENTS .	Specifies the number of segments into which the image was partitioned for the error containment purposes. For ICER compression, the data within each segment is compressed independently, so that data loss across segments is compartmentalized or contained across segments. For PHX, used for JPEG only.	"1" to 64"	**SOURCE** seg_total **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer **UNITS** n/a
INST_CMPRS_SEGMENT_STATUS	Specifies a bit mask which provides the status of decoding the nth segment. For PHX, used for JPEG only.	"0" = failure "1" = success	**SOURCE** Returned from JPEG decompression routine **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** string **UNITS** n/a
INST_CMPRS_SEG_FIRST_LINE	Specifies an array of values which each nth element identifies the line within a source image that corresponds to the first line the nth compression segment applies. Value of "-1" denotes the indeterminate case when decompressor cannot process the segment, or cannot determine seq_quality for a segment that it could decompress. For PHX, used for JPEG only.	"-1" to "1024"	**SOURCE** Extracted from JPEG segment **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer array **UNITS** n/a
INST_CMPRS_SEG_FIRST_LINE_SAMP	Specifies an array of values which each nth element identifies the line sample within a source image that corresponds to the first line sample the nth compression segment applies.	"-1" to "1024"	**SOURCE** Extracted from JPEG segment **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer array **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Value of "-1" denotes the indeterminate case when decompressor cannot process the segment, or cannot determine seq_quality for a segment that it could decompress. For PHX, used for JPEG only.			
INST_CMPRS_SEG_LINES	Specifies an array of elements in which the nth element identifies the total number of data instances along the vertical axis the nth compression segment defines. Value of "-1" denotes the indeterminate case when decompressor cannot process the segment, or cannot determine seq_quality for a segment that it could decompress. For PHX, JPEG image segment height.	"-1" to "1024"	**SOURCE** comp_segh **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer array **UNITS** n/a
INST_CMPRS_SEG_MISSING_PIXELS	Specifies an array of elements in which the nth element identifies the total number missing pixels that the nth compression segment defines.		**SOURCE** • If frame syncing fails: comp_segh * comp_segw • Otherwise, "0". **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer array **UNITS** n/a
INST_CMPRS_SEG_SAMPLES	Specifies an array of elements in which the nth element identifies the total number of data instances along the horzontal axis the nth compression segment defines. Value of "-1" denotes the indeterminate case when decompressor cannot process the segment, or cannot determine seq_quality for a segment that it could decompress. For PHX, JPEG image segment width.	"-1" to "1024"	**SOURCE** comp_segw **LOCATION** COMPRESSION_PARMS (Group)	**TYPE** integer array **UNITS** n/a
INTERCHANGE_FORMAT	Specifies the manner in which data items are stored.	"ASCII", "BINARY"	**SOURCE** Static Value **LOCATION** • IMAGE_HEADER (Object) • IMAGE (Object)	**TYPE** string(6) **UNITS** n/a
INVALID_CONSTANT	Specifies the value used when the received data are out of the legitimate range of values.	<u>Most OPGS-gen'd Products</u> "0.0"	**SOURCE** Static Value	**TYPE** float or float array

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	For PHX, the value should be 0.0 for most MIPL-generated products.	XYZ "(0.0, 0.0, 0.0)" Surface Normal (UVW) "(0.0, 0.0, 0.0)"	**LOCATION** IMAGE (Object)	**UNITS** n/a
LABEL_RECORDS	Specifies the number of physical file records that contain only (PDS) label information. The number of data records in a file is determined by subtracting the value of label_records from the value of file_records. Note: In the PDS, the use of label_records along with other file-related data elements is fully described in the Standards Reference.	"0" to n	**SOURCE** Calculated by size of PDS label **LOCATION** FILE (Class)	**TYPE** integer **UNITS** n/a
LED_BIT_MASK LED_BITMASK [V]	Specifies the status of MECA-OM LED's as a 16-bit unsigned integer, one bit per LED. It is represented in the label as a HEX value, with the 4 most significant bits unused (they are zero's). For example, the 16-bit binary Bitmask with Hex value C80 ("16#0000C80#'") is: Hex: (unused) (C) (8) (0) Binary: 0000 1100 1000 0000 Reading the Bit positions across the 4 binary nibbles in the Bitmask from right (lsb) to left (msb), each "1" denotes an "ON" switch of the corresponding LED. So, this translates to filters Red #3, Red #2 and Red #1 as all being ON.	"16#0000<Hex string>#" where, <Hex string> = 3 Hex values, 1 for each of the 3 least significant (rightmost) 4-bit nibbles in the 16-bit Bitmask. The most significant 4^{th} nibble (leftmost) is unused (all zeros). and given, **Hex-to-Binary Conversion** / **LED Mapped to Bit Position** Hex Binary / LED Bit # "0" 0000 / Green #3 0 "1" 0001 / Blue #3 1 "2" 0010 / Blue #2 2 "3" 0011 / Green #2 3 "4" 0100 / Green #1 4 "5" 0101 / Blue #1 5 "6" 0110 / UV #3 6 "7" 0111 / Red #3 7 "8" 1000 / UV #2 8 "9" 1001 / UV #1 9 "A" 1010 / Red #2 10 "B" 1011 / Red #1 11 "C" 1100 "D" 1101 "E" 1110	**SOURCE** led_bitmask **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** Hex **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"F" 1111		
LIGHT_SOURCE_DISTANCE	Specifies the distance from the target body center and secondary light source center.		**SOURCE** GSE software **LOCATION** GROUND_SUPPORT_EQUIPMENT (Group)	**TYPE** float **UNITS** meters (<m> unit tag required)
LIGHT_SOURCE_NAME	Specifies the name of the light source used in observations when it is not the Sun.		**SOURCE** GSE software **LOCATION** GROUND_SUPPORT_EQUIPMENT (Group)	**TYPE** string **UNITS** n/a
LIGHT_SOURCE_TYPE	Specifies the source of illumination used in instrument calibration.		**SOURCE** GSE software **LOCATION** GROUND_SUPPORT_EQUIPMENT (Group)	**TYPE** string **UNITS** n/a
LINES	Specifies the total number of data instances along the vertical axis of an image. Note: In PDS label convention, the number of lines is stored in a 32-bit integer field. The minimum value of 0 indicates no data received.	EDRs "1" to "1024" RDRs variable beyond 1024	**SOURCE** (subframe_y_end - subframe_y_start) + 1 **LOCATION** IMAGE (Object)	**TYPE** integer **UNITS** n/a
LINE_CAMERA_MODEL_OFFSET	Specifies the location of the image origin with respect to the camera model's origin. For CAHV/CAHVOR models, this origin is not the center of the camera, but is the upper-left corner of the "standard"-size image, which is encoded in the CAHV vectors. (MIPL Projection - Perspective)		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixels (<pixel> unit tag required)
LINE_PROJECTION_OFFSET	Specifies the line coordinate of the location in the image of the "special" point of the mosaic. For Polar projections, this is the nadir of the polar projection. For Vertical and Orthographic projections, this is the origin of the projected coordinate system grid (i.e., X=0.0, Y=0.0). Not applicable to other projections.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixels (<pixel> unit tag required)
LINE_SAMPLES	Specifies the total number of data instances along the horizontal axis of an image.	EDRs "1" to "1024" RDRs variable beyond 1024	**SOURCE** (subframe_x_end - subframe_x_start) + 1 **LOCATION** IMAGE (Object)	**TYPE** integer **UNITS** n/a
LOCAL_TRUE_SOLAR_TIME	Specifies the local true solar time, or LTST. It is one of two types of solar time used to	NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	**SOURCE** Calculated from SPICE using:	**TYPE** string(12)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and areocentric 'right ascension' and 'declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet's surface. Specifically, LTST = (a(P) - a(TS)) * (24 / 360) + 12 where, LTST = the local true solar time in true solar hours a(P) = ARA of the point on the planet's surface in deg a(TS) = ARA of the true sun in deg The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours,		• SCLK kernel • P kernel • Landing Site kernel **LOCATION** IDENTIFICATION (Class)	**UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours is added to the difference in ARAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the ARA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0. The use of 'true solar' time units can be extended to define a true solar day as 24 true solar hours. Due to the eccentricity of planetary orbits and the inclination of orbital planes to equatorial planes (obliquity), the Sun does not move at a uniform rate over the course of a planetary year. Consequently, the number of SI seconds in a true solar day, hour, minute or second is not constant. See also LOCAL_MEAN_SOLAR_TIME. This element replaces the older MPF_LOCAL_TIME, which should no longer be used.			
MAP_PROJECTION_TYPE	Specifies the type of projection characteristic of a given map. When in a SURFACE_PROJECTION group, defines the surface-based map projection used in the image.	"CYLINDRICAL", "VERTICAL", "PERSPECTIVE", "POLAR", "ORTHOGRAPHIC", "CYLINDRICAL-PERSPECITIVE"	**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** string **UNITS** n/a
MAP_RESOLUTION	Specifies the scale of a given map. Please refer to the definition for MAP_SCALE for a more complete definition. When in a SURFACE_PROJECTION group, defines the resolution of the map in pixels/degree. For CYLINDRICAL, this is constant throughout. For POLAR, this is for the Elevation (radial) direction only. For PERSPECTIVE and CYLINDRICAL-PERSPECITIVE, this is at the center of the output camera model. Not applicable to VERTICAL. Note: MAP_RESOLUTION and MAP_SCALE both define the scale of a		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float array[2] **UNITS** pixels (<pixel> unit tag required)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	map except that they are expressed in different units: MAP_RESOLUTION is in pixels/deg and MAP_SCALE is in meters/pixel. If two values are present, the first measures in the line direction while the second measures in the sample direction.			
MAP_SCALE	Specifies the scale of a given map. The scale is defined as the ratio of the actual distance between two points on the surface of the target body to the distance between the corresponding points on the map. MAP_SCALE references the scale of a map at a certain reference point or line. Certain map projections vary in scale throughout the map. When in a SURFACE_PROJECTION group, defines the scale of the map in meters/pixel. Applicable to VERTICAL and ORTHOGRAPHIC projections only. Note: MAP_RESOLUTION and MAP_SCALE both define the scale of a map except that they are expressed in different units: MAP_RESOLUTION is in pixels/deg and MAP_SCALE is in meters/pixel. If two values are present, the first measures in the line direction while the second measures in the sample direction.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float array[2] **UNITS** m/pixel (<m/pixel> unit tag required)
MAXIMUM	Specifies the largest value occurring in a given instance of the data object. Note: For PDS applications -- because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e., anywhere between an 'OBJECT =' and an 'END_OBJECT'.		**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** float **UNITS** n/a
MAXIMUM_ELEVATION	Specifies the elevation (as defined by the coordinate system) of the first line of the image. Applies to MIPL projections Cylindrical, Perspective and Cylindrical-Perspective.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
				tag required)
MEAN	Specifies the average of the DN values in the image array.		**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** float **UNITS** n/a
MEDIAN	Specifies the median value (middle value) occurring in a given instance of the data object. Because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e., anywhere between an 'OBJECT =' and an 'END OBJECT'.		**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** float **UNITS** n/a
MINIMUM	Specifies the smallest value occurring in a given instance of the data object. Note: For PDS and Mars Observer applications -- because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e., anywhere between an 'OBJECT =' and an 'END_OBJECT'.		**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** float **UNITS** n/a
MINIMUM_ELEVATION	Specifies the elevation (as defined by the coordinate system) of the last line of the image for Cylindrical map projections. Applies to mosaics witth Cylindrical, Perspective and Cylindrical-Perspective projections.	-180 to +180	**SOURCE** Calculation **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
MISSING_CONSTANT	Specifies the flag value used in the image to indicate that no science data are available for any given pixel. See the specific product definitions for standard values used for each product.	"0.0" <u>XYZ</u> "(0.0, 0.0, 0.0)" <u>Surface Normal (UVW)</u> "(0.0, 0.0, 0.0)"	**SOURCE** Static Value **LOCATION** IMAGE (Object)	**TYPE** float or float array **UNITS** n/a
MISSION_NAME	Specifies a major planetary mission or project. A given planetary mission may be associated with one or more spacecraft. Note that mosaics may contain more than one value in an array.	"PHOENIX LANDER"	**SOURCE** Static Value **LOCATION** IDENTIFICATION (Class)	**TYPE** string array **UNITS** n/a
MISSION_PHASE_NAME	Specifies the commonly-used identifier of a mission phase.	"CRUISE", "PRIMARY MISSION", "EXTENDED MISSION"	**SOURCE** Operator Supplied Parameter **LOCATION** IDENTIFICATION (Class)	**TYPE** string(30) **UNITS** n/a
MODEL_COMPONENT_1	Specifies a set of values representing the		**SOURCE**	**TYPE**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	first component of a model. The significance (or meaning) of this array of values is indicated by the first value of the MODEL_COMPONENT_ID and/or MODEL_COMPONENT_NAME elements. The interpretation of the values themselves depends on the model but they commonly represent a vector, a set of polynomial coefficients, or a simple numeric parameter. For example, for a geometric camera model with a value of "CAHV" for MODEL_TYPE, the first value of the MODEL_COMPONENT_NAME data element is CENTER, meaning that the MODEL_COMPONENT_1 is a focal center vector. The three items in this vector provide X, Y, and Z coordinates of the focal point of the camera. The exact details about each model component vector are provided in MODEL_DESC.		Calculation **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	float array **UNITS** n/a
MODEL_COMPONENT_2	Specifies the value of the component of the MODEL_COMPONENT_ID for the second element.		**SOURCE** Calculation **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** float array **UNITS** n/a
MODEL_COMPONENT_3	Specifies the value of the component of the MODEL_COMPONENT_ID for the third element.		**SOURCE** Calculation **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** float array **UNITS** n/a
MODEL_COMPONENT_4	Specifies the value of the component of the MODEL_COMPONENT_ID for the fourth element.		**SOURCE** Calculation **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** float array **UNITS** n/a
MODEL_COMPONENT_5	Specifies the value of the component of the MODEL_COMPONENT_ID for the fifth element.		**SOURCE** Calculation **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** float array **UNITS** n/a
MODEL_COMPONENT_6	Specifies the value of the component of the MODEL_COMPONENT_ID for the sixth element.		**SOURCE** Calculation **LOCATION**	**TYPE** float array **UNITS**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
			GEOMETRIC_CAMERA_MODEL (Group)	n/a
MODEL_COMPONENT_ID	Specifies a sequence of identifiers (usually 1 character), where each identifier corresponds to a model component vector. It is used in conjunction with the MODEL_COMPONENT_n elements, where "n" is a number. The first id in the sequence corresponds to MODEL_COMPONENT_1, the second corresponds to MODEL_COMPONENT_2, etc. For example, for a geometric camera model with a value of "CAHV" for MODEL_TYPE, the MODEL_COMPONENT_ID would be (C, A, H, V). Please see the MODEL_COMPONENT_NAME data element for more details.	0 = "NONE" 1 = "(C,A,H,V)" 2 = "(C,A,H,V,O,R)" 3 = "(C,A,H,V,O,R,E,T,P)"	**SOURCE** Table Lookup **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** string array **UNITS** n/a
MODEL_COMPONENT_NAME	Specifies a sequence of names, where each name identifies its corresponding model component vector. It is used in conjunction with the MODEL_COMPONENT_n elements, where "n" is a number. The first name in the sequence identifies MODEL_COMPONENT_1, the second identifies the MODEL_COMPONENT_2, etc. For example, for a geometric camera model with a value of "CAHV" for MODEL_TYPE, the MODEL_COMPONENT_NAME would be (CENTER, AXIS, HORIZONTAL, VERTICAL). The three values of MODEL_COMPONENT_1 would describe the focal center vector; the three values of MODEL_COMPONENT_2 would describe the pointing direction (axis) vector; the three values of MODEL_COMPONENT_3 would describe the horizontal image plane vector, and the three values of the MODEL_COMPONENT_4 would describe the vertical image plane vector.	0 = NONE 1 = ("CENTER", "AXIS", "HORIZONTAL", "VERTICAL") 2 = ("CENTER", "AXIS", "HORIZONTAL", "VERTICAL", "OPTICAL", "RADIAL") 3 = ("CENTER", "AXIS", "HORIZONTAL", "VERTICAL", "OPTICAL", "RADIAL", "ENTRANCE", "MTYPE", "MPARM")	**SOURCE** Table Lookup **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** string array **UNITS** n/a
MODEL_TYPE	Specifies an identifier for the type or kind of model. The value should be one of a well defined set, providing an application program with sufficient information to know how to handle the rest of the parameters	0 = "NONE" 1 = "CAHV" 2 = "CAHVOR" 3 = "CAHVORE"	**SOURCE** Table Lookup **LOCATION** GEOMETRIC_CAMERA_MODEL (Group)	**TYPE** string(63) **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	within the model. (CAHVORE is the only one that uses model component vectors 1-9.) For details on the definitions of the valid camera model types, see [Ref 11] through [Ref 16].			
OFFSET_MODE_ID	Specifies the analog value that is subtracted from the video signal prior to the analog/digital converters. For PHX, the image A/D (analog-to-digital) conversion offset for SSI only.	"0" to "4095"	**SOURCE** ccd_conv_offset **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** n/a
OPS_TOKEN	A 32-bit operations token represented as a 8-digit Hex value.		**SOURCE** Telemetry instrument header **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** hexidecimal **UNITS** n/a
OPS_TOKEN_ACTIVITY	A 16-bit Activity identifier extracted from the 16 most significant bits of the 32-bit operations token. Represented as a 4-digit Hex value.		**SOURCE** Telemetry instrument header **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** hexidecimal **UNITS** n/a
OPS_TOKEN_COMMAND	A 12-bit Command identifier extracted from the 12 least significant bits of the 32-bit operations token. Represented as a decimal value.		**SOURCE** Telemetry instrument header **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** hexidecimal **UNITS** n/a
OPS_TOKEN_PAYLOAD	A 4-bit Payload identifier extracted from the 4 bits bookended by the Activity and Command bits in the 32-bit operations token. Represented as a single-digit Hex value.		**SOURCE** Telemetry instrument header **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** hexidecimal **UNITS** n/a
ORIGIN_OFFSET_VECTOR	Specifies the offset from the reference coordinate system's origin to the origin of the coordinate system being defined by the enclosing COORDINATE_SYSTEM group. In other words, it is the location of the current system's origin as measured in the reference system. For PHX, here is an example: In the case of the PAYLOAD_COORDINATE_SYSTEM group, ORIGIN_OFFSET_VECTOR specifies where the origin of the Payload Frame as measured in the		**SOURCE** Calculation **LOCATION** Group Dependent: • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group)	**TYPE** float array[3] **UNITS** meters

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Reference (Site) Frame.			
ORIGIN_ROTATION_QUATERNION	Specifies an array of four values that specifies the rotation of the coordinate system being defined by the enclosing COORDINATE_SYSTEM group, relative to the reference system. Mathematically this can be expressed as follows: Given a vector expressed in the current frame, multiplication by this quaternion will give the same vector as expressed in the reference frame. Quaternions are expressed as a set of four numbers in the order: (s, v1, v2, v3) where, s = cos(theta/2) v(n) = sin(theta/2)*a(n). theta = the angle of rotation a = (x,y,z) vector around which the rotation occurs. For PHX, the value for ORIGIN_ROTATION_QUATERNION that defines a coordinate frame like Payload Frame is computed with respect to only the orientations of the frame's axes… regardless of whether POSITIVE_ELEVATION_DIRECTION is declared to be “UP” or “DOWN”. For PHX, here is an example: In the case of the PAYLOAD_COORDINATE_SYSTEM group, ORIGIN_ROTATION_QUATERNION describes the rotation of the Lander (about the ORIGIN_OFFSET_VECTOR) relative to the Payload Frame.		**SOURCE** Calculation **LOCATION** Group Dependent: • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group)	**TYPE** float array[4] **UNITS** n/a
PACKET_MAP_MASK	Specifies a binary or hexadecimal number identifying which of a data file's expected packets were actually received. The digits correspond positionally with the relative packet numbers of the data file. The bits are to be read left to right; i.e., the first (left-most) digit of the number corresponds to the first packet of the data file. A bit value of 1 indicates that the packet was received; a value of 0 indicates that it was not received.		**SOURCE** Calculation **LOCATION** TELEMETRY (Class)	**TYPE** non-decimal **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	The number is stored in the PDS radix notation of <radix>#<value>#.			
PDS_VERSION_ID	Specifies the version number of the PDS standards document that is valid when a data product label is created. Values for the PDS_version_id are formed by appending the integer for the latest version number to the letters 'PDS'. Examples: PDS3, PDS4.	"PDS3"	**SOURCE** PDS **LOCATION** PDS required	**TYPE** string(6) **UNITS** n/a
PIXEL_AVERAGING_HEIGHT	Specifies the vertical dimension, in pixels, of the area over which pixels were averaged prior to image compression.		**SOURCE** • image_dec_mode • image_dec_mode_y • DOWNSAMPLE_METHOD NOTES: • SSI - If DOWNSAMPLE_METHOD is "NONE" or "HARDWARE", then value is "1". - If DOWNSAMPLE_METHOD is "SOFTWARE", then value is equal to image_dec_mode. - If DOWNSAMPLE_METHOD is "BOTH", then value is "4". - Otherwise, value is "1". • RAC and OM - If DOWNSAMPLE_METHOD is "NONE", then value is "1". - If DOWNSAMPLE_METHOD is "SOFTWARE", then value is equal to image_dec_mode. **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** integer **UNITS** pixel
PIXEL_AVERAGING_WIDTH	Specifies the horizontal dimension, in pixels, of the area over which pixels were averaged prior to image compression.		**SOURCE** • image_dec_mode • image_dec_mode_x • DOWNSAMPLE_METHOD NOTES: • SSI - If DOWNSAMPLE_METHOD is "HARDWARE", then value is "4". - If DOWNSAMPLE_METHOD is "NONE", then value is "1". - If DOWNSAMPLE_METHOD is "SOFTWARE", then value is equal to	**TYPE** integer **UNITS** pixel

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
			image_dec_mode. - If DOWNSAMPLE_METHOD is "BOTH", then value is "4". • RAC and OM - If DOWNSAMPLE_METHOD is "NONE", then value is "1". - If DOWNSAMPLE_METHOD is "SOFTWARE", then value is equal to image_dec_mode. **LOCATION** INSTRUMENT_STATE_PARMS (Group)	
PIXEL_DOWNSAMPLE_OPTION	Specifies whether to downsample the image(s), and if so, which pixel resolution downsample method to use. "SW" - Downsampling done in software via sampling upper right pixel. "SW_MEAN" – Downsampling done in software by alculaiton of the mean. "HW" – Binning done in hardware. "HWSW" – Binning done in hardware followed by software downsampling via sampling upper right pixel. "HWSW_MEAN" – Binning done in hardware followed by software downsampling by calculation of the mean.	0 = "NONE" 1 = "SW" 2 = "SW_MEAN" 3 = "HW" 4 = "HWSW" 5 = "HWSW_MEAN"	**SOURCE** • image_dec_avg • DOWNSAMPLE_METHOD NOTES: • SSI - If DOWNSAMPLE_METHOD is "HARDWARE", then value is "HW". - If DOWNSAMPLE_METHOD is "NONE", then value is "None". - If DOWNSAMPLE_METHOD is "SOFTWARE" and image_dec_avg is "0", then value is "SW". - If DOWNSAMPLE_METHOD is "SOFTWARE" and image_dec_avg is NOT "0", then value is "SW Mean". - If DOWNSAMPLE_METHOD is "BOTH" and image_dec_avg is "0", then value is "HWSW". - If DOWNSAMPLE_METHOD is "BOTH" and image_dec_avg is NOT "0", then value is "HWSW_Mean". • RAC and OM - If DOWNSAMPLE_METHOD is "NONE", then value is "None". - If DOWNSAMPLE_METHOD is "SOFTWARE" and image_dec_avg is "0", then value is "SW". - If DOWNSAMPLE_METHOD is "SOFTWARE" and image_dec_avg is NOT "0", then value is "SW_Mean". **LOCATION** INSTRUMENT_STATE_RESULTS (Group)	**TYPE** string **UNITS** n/a
PLANET_DAY_NUMBER	Specifies the number of sidereal days	NOTE: Value will be uncalibrated if	**SOURCE**	**TYPE**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	(rotation of 360 degrees) elapsed since a reference day (e.g., the day on which a landing vehicle set down). Days are measured in rotations of the planet in question from the reference day. For PHX, the reference day is “1”, as Landing day is Sol 1.	SPICE kernels are unavailable.	Calculation **LOCATION** IDENTIFICATION (Class)	integer **UNITS** n/a
POSITIVE_AZIMUTH_DIRECTION	Specifies the direction in which azimuth is measured in positive degrees for an observer on the surface of a body. The azimuth is measured with respect to the elevational reference plane. A value of CW indicates that Azimuth is measured positively Clockwise, and CCW indicates that Azimuth increases positively Counter-clockwise.	“CLOCKWISE””	**SOURCE** Static Value determined by coordinate frame definitions. **LOCATION** • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group)	**TYPE** string **UNITS** n/a
POSITIVE_ELEVATION_DIRECTION	Specifies the direction in which elevation is measured in positive degrees for an observer on the surface of a body. The elevation is measured with respect to the azimuthal reference plane. A value of “UP” indicates that elevation is measured positively upwards, i.e., the zenith point would be at +90 degrees and the nadir point at -90 degrees. “DOWN” indicates that the elevation is measured positively downwards; the zenith point would be at -90 degrees and the nadir point at +90 degrees. For the PHX operational coordinate frames, which follow the Mars Pathfinder convention, increasing elevation (“UP”) moves towards the negative Z axis.	“UP”	**SOURCE** Static Value determined by coordinate frame definitions. **LOCATION** • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group)	**TYPE** string **UNITS** n/a
PROCESSING_HISTORY_TEXT	Specifies an entry for each processing step and program used in generating a particular data file.	“CODMAC LEVEL 1 TO LEVEL 2 CONVERSION VIA JPL/MIPL PHXTELEMPROC”	**SOURCE** Static Value **LOCATION** HISTORY (Class)	**TYPE** string **UNITS** n/a
PROCESSING_INFORMATION_TEXT	Specifies information about the processing used to generate higher level products that is not covered by other PDS label keywords.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string array **UNITS** n/a
PRODUCER_INSTITUTION_NAME	Specifies the identity of a university,	“MULTIMISSION IMAGE	**SOURCE**	**TYPE**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	research center, NASA center or other institution associated with the production of a data set. This would generally be an institution associated with the element PRODUCER_FULL_NAME.	PROCESSING SUBSYSTEM, JET PROPULSION LAB"	Static Value **LOCATION** IDENTIFICATION (Class)	string(60) **UNITS** n/a
PRODUCT_CREATION_TIME	Specifies the UTC system format for the time when a product was created.	YYYY-MM-DDThh:mm:ss[.fff]Z	**SOURCE** Calculation **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a
PRODUCT_ID	Specifies a permanent, unique identifier assigned to a data product by its producer. For PHX, it is the filename minus the extension. Note: In the PDS, the value assigned to product_id must be unique within its data set. Additional note: The PRODUCT_ID can describe the lowest-level data object that has a PDS label.		**SOURCE** Filename minus the extension. **LOCATION** IDENTIFICATION (Class)	**TYPE** string(40) **UNITS** n/a
PRODUCT_VERSION_ID	Specifies the version of an individual product within a data set. PRODUCT_VERSION_ID is intended for use within AMMOS to identify separate iterations of a given product, which will also have a unique FILE_NAME. For PHX, PRODUCT_VERSION_ID includes a Version field that begins with "V" followed by the Version decimal number of the controlling SIS document. Example: "V2.0 D-22846" NOTE: This might not be the same as the data set version that is an element of the DATA_SET_ID value.	"V<*vernum*> D-<*vernum*>"	**SOURCE** Static Value **LOCATION** IDENTIFICATION (Class)	**TYPE** string(12) **UNITS** n/a
PROJECTION_AXIS_OFFSET	Specifies the radius of a circle, where the circle represents the rotation around the projection origin by the synthetic or fictitious camera used to calculate each column in the Cylindrical-Perspective projection. The radius is the distance from the camera to		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** meters (<m> unit tag

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	the origin. If the value is positive, the fictitious camera is to the right of the origin when seen from behind (i.e. in the direction of its boresight). If negative, the camera is on the left. If the keyword does not appear, it is assumed to be 0, i.e. the camera rotates in place without describing a circle.			required)
PROJECTION_AZIMUTH	Specifies the azimuth, in degrees, of the horizontal center of projection for the PERSPECTIVE projection (loosely, where the camera model is pointing). For the Cylindrical-Perspective projection, it defines the angle at which the synthetic camera for each column is rotated relative to the vector tangent to the circle described by the camera center (see PROJECTION_AXIS_OFFSET). This is used to model toe-in of the synthetic camera. A positive value rotates the camera counterclockwise when seen from above, so PROJECTION_AZIMUTH and PROJECTION_AXIS_OFFSET share the same sign then the synthetic camera is toed in. Absence of the keyword does not indicate the azimuth (toe-in) is zero. Instead, the camera model group implicitly defines the amount of toe-in via the A vector.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
PROJECTION_ELEVATION	Specifies the elevation, in degrees, of the vertical center of projection (loosely, where the camera is pointing). For PERSPECTIVE, this applies to the single output camera model; for CYLINDRICAL-PERSPECTIVE it applies to each column's output camera model.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
PROJECTION_ELEVATION_LINE	Specifies the image line which corresponds to PROJECTION_ELEVATION for each column of the CYLINDRICAL-PERSPECTIVE projection.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixels (<pixel> unit tag required)
PROJECTION_ORIGIN_VECTOR	Specifies the location of origin of the projection. This is an xyz point from which all the azimuth/elevation rays emanate.		**SOURCE** Image Processing Software	**TYPE** float array[3]

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Applies to Polar, Cylindrical, Cylindrical-Perspective and Orthographic projections. For the Cylindrical-Perspective projection, this is the point around which the synthetic camera orbits. If the value is not present, it can be derived from the C vector of the camera model group (in which case the PROJECTION_AXIS_OFFSET should be 0). For the Orthographic projection, the PROJECTION_ORIGIN_VECTOR defines a point in a plane normal to the direction of projection (given by PROJECTION_Z_AXIS_VECTOR) that serves as the origin. All points that lie on a line through the PROJECTION_ORIGIN_VECTOR in the direction of the PROJECTION_Z_AXIS_VECTOR will be located at X=Y=0 in the orthographic projection.		**LOCATION** SURFACE_PROJECTION_PARMS (Group)	**UNITS** meters (<m> unit tag required)
ORTHO_PROJ_X_AXIS_VECTOR PROJECTION_X_AXIS_VECTOR [V]	Specifies a unit vector giving the direction of the X-axis lying within the plane of projection for the orthographic projection. Note: Required if and only if MAP_PROJECTION_TYPE = "ORTHOGRAPHIC". Note:This is only one of several equivalent ways that the orientation of the orthographic projection might be specified (others are by a 3x3 rotation matrix with rows equal to these vectors, as three Euler angles, or as a unit quaternion).		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float array[3] **UNITS** meters (<m> unit tag required)
ORTHO_PROJ_Y_AXIS_VECTOR PROJECTION_Y_AXIS_VECTOR [V]	Specifies a unit vector giving the direction of the Y-axis lying within the plane of projection for the orthographic projection. Note: Required if and only if MAP_PROJECTION_TYPE = "ORTHOGRAPHIC". Note:This is only one of several equivalent ways that the orientation of the		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float array[3] **UNITS** meters (<m> unit tag required)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	orthographic projection might be specified (others are by a 3x3 rotation matrix with rows equal to these vectors, as three Euler angles, or as a unit quaternion).			
ORTHO_PROJ_Z_AXIS_VECTOR PROJECTION_Z_AXIS_VECTOR [V]	Specifies a unit vector giving the direction of the Z-axis lying within the plane of projection for the orthographic projection. Note: Required if and only if MAP_PROJECTION_TYPE = "ORTHOGRAPHIC". Note:This is only one of several equivalent ways that the orientation of the orthographic projection might be specified (others are by a 3x3 rotation matrix with rows equal to these vectors, as three Euler angles, or as a unit quaternion).		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float array[3] **UNITS** meters (<m> unit tag required)
IMAGE_RADIANCE_FACTOR RADIANCE_FACTOR_PER_IMAGE [V]	For a multi-input image (such as a mosaic), specifies the overall multiplicative factor that was applied to each image as the mosaic was being assembled. Together with RADIANCE_OFFSET_PER_IMAGE, this specifies a simple linear adjustment to each input image that can be used for (limited) radiometric seam correction. Each pixel within a given input image receives the same correction. The order of images is as defined in the input list file (generally delivered along with the mosaic). Note: This correction is applied after radiometric correction, if any, performed on the inputs.		**SOURCE** Image Processing Software NOTES: • Output Pixel DN = 0 if Input Pixel DN = 0. • Otherwise, Output Pixel DN = Input Pixel DN * RADIANCE_FACTOR_PER_IMAGE(n) + RADIANCE_OFFSET_PER_IMAGE(n) **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** float array **UNITS** n/a
RADIANCE_OFFSET	Specifies the constant value by which a stored radiance is added. Note: Expressed as an equation: true_radiance_value = radiance_offset + radiance_scaling_factor*stored_radiance value. There are 2 types of radiometric corrections, with a 3rd one optional: <u>Radiance-calibrated RDRs</u> In SSI case, these "RAD" (and "RAL") RDRs have been scaled to absolute		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** float **UNITS** • For RAD: WATT* M**-2* NM**-1* SR**-1 • For IOF: Unitless I/F • For CCD:

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	radiance units using either pre-flight radiometric calibration coefficients or calibration coefficients derived from in-flight observations of the SSI calibration target. The units on these files are (W/m^2/nm/sr). Radiance factor-calibrated RDRs These "IOF" (and "IOL") RDRs are unitless but have values in the range of 0.0 to 1.0 (for example, average bright Mars soils exhibit I/F ~ 0.35 at 750 nm and I/F ~ 0.05 at 410 nm), and greater than 1.0 in certain conditions (near Sun sky imaging). Instrumentally-calibrated RDRs (optional) As a future option, these "CCD" (and "CCL") RDRs would have had no radiance scaling applied, so the units on these files would be "corrected" DN.			DN NOTE: Ask Mark about the units
RADIANCE_OFFSET_PER_IMAGE	For a multi-input image (such as a mosaic), specifies the overall additive offset that was applied to each image as the mosaic was being assembled. Together with RADIANCE_FACTOR_PER_IMAGE, this specifies a simple linear adjustment to each input image that can be used for (limited) radiometric seam correction. Each pixel within a given input image receives the same correction. Input values of 0 are not adjusted. The order of images is as defined in the input list file (generally delivered along with the mosaic). Note: This correction is applied after radiometric correction, if any, performed on the inputs.		**SOURCE** Image Processing Software NOTES: • Output Pixel DN = 0 if Input Pixel DN = 0. • Otherwise, Output Pixel DN = Input Pixel DN * RADIANCE_FACTOR_PER_IMAGE(n) + RADIANCE_OFFSET_PER_IMAGE(n) **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** float array **UNITS** DN
RADIANCE_SCALING_FACTOR	Specifies the constant value by which a stored radiance is multiplied. Note: Expressed as an equation: true_radiance_value = radiance_offset + radiance_scaling_factor*stored_radiance value There are 2 types of radiometric corrections, with a 3rd one optional:		**SOURCE** Image Processing Software **LOCATION** • DERIVED_IMAGE_PARMS (Group) • IMAGE (Object)	**TYPE** float **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Radiance-calibrated RDRs In SSI case, these "RAD" (and "RAL") RDRs have been scaled to absolute radiance units using either pre-flight radiometric calibration coefficients or calibration coefficients derived from in-flight observations of the SSI calibration target. The units on these files are (W/m^2/nm/sr). Radiance factor-calibrated RDRs These "IOF" (and "IOL") RDRs are unitless but have values in the range of 0.0 to 1.0 (for example, average bright Mars soils exhibit I/F ~ 0.35 at 750 nm and I/F ~ 0.05 at 410 nm), and greater than 1.0 in certain conditions (near Sun sky imaging). Instrumentally-calibrated RDRs (optional) As a future option, these "CCD" (and "CCL") RDRs would have had no radiance scaling applied, so the units on these files would be "corrected" DN.			
RADIOMETRIC_CORRECTION_TYPE	Identifies the method used for radiometric correction. Values include "TAMCAL" and "RACCAL" for corrections performed by the science team, and "MIPLRAD", "MIPLRAD2", "MIPLRAD3" for corrections performed by MIPL (the difference being the dark current model used). "NONE" indicates no radiometric correction has been performed.	"TAMCAL", "RACCAL", "MIPLRAD", "MIPLRAD2", "MIPLRAD3", "NONE"	**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** string **UNITS** n/a
RANGE_ORIGIN_VECTOR	Specifies the 3-D space from which the Range values are measured in a Range RDR. This will normally be the same as the C point of the camera. It is expressed in the coordinate system specified by the REFERENCE_COORD_SYSTEM_* keywords in the enclosing DERIVED_IMAGE_PARMS group.		**SOURCE** Calculation **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** float array[3] **UNITS** n/a
RECEIVED_PACKETS	Specifies the total number of telemetry packets which constitute a reconstructed data product.		**SOURCE** Calculation **LOCATION** TELEMETRY (Class)	**TYPE** integer **UNITS** n/a
RECORD_BYTES	Specifies the number of bytes in a physical file record, including record terminators and separators.	"0" to n	**SOURCE** Calculation	**TYPE** integer

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.		**LOCATION** FILE (Class)	**UNITS** n/a
RECORD_TYPE	Specifies the record format of a file. Note: In the PDS, when record_type is used in a detached label file it always describes its corresponding detached data file, not the label file itself. The use of record_type along with other file-related data elements is fully described in the PDS Standards Reference.	"FIXED_LENGTH"	**SOURCE** Static Value **LOCATION** FILE (Class)	**TYPE** string(20) **UNITS** n/a
REFERENCE_AZIMUTH	Specifies the azimuth of the line extending from the center of the image to the top center of the image with respect to a polar projection.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
REFERENCE_COORD_SYSTEM_INDEX	Specifies which instance of the coordinate system named by REFERENCE_COORD_SYSTEM_NAME is the reference coordinate system for the group in which the keyword occurs. This index is a set of integers which serve to identify coordinate system instances in a mission-specific manner. NOTE: For PHX, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the lander may potentially have moved, e.g. due to arm motion. The full counter may have up to 2 values (SITE, POSITION), but normally only the first value (for SITE frames) or the first two values (for LOCAL_LEVEL or PAYLOAD frames) are used for defining coordinate system instances. It is legal to use any number of indices to describe a coordinate system instance, however. See also REFERENCE_COORD_SYSTEM_NAME and COORDINATE_SYSTEM_INDEX.		**SOURCE** Group dependent: • Calculation **LOCATION** Group dependent: a) GEOMETRIC_CAMERA_MODEL (Group) b) PAYLOAD_COORDINATE_SYSTEM (Group) c) LOCAL_LEVEL_COORDINATE_SYSTEM (Group) d) SITE_COORDINATE_SYSTEM (Group) e) PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) f) SITE_DERIVED_GEOMETRY_PARMS (Group) g) DERIVED_IMAGE_PARMS (Group) h) SURFACE_PROJECTION_PARMS (Group) i) SURFACE_MODEL_PARMS (Group)	**TYPE** integer array[2] **UNITS** n/a
REFERENCE_COORD_SYSTEM_NAME	Specifies the full name of the reference coordinate system for the group in which	EDRs ""SITE_FRAME",	**SOURCE** Software dependent	**TYPE** string(20)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	the keyword occurs. All vectors and positions relating to 3-D space within the enclosing group are expressed using this reference coordinate system. Non-unique coordinate systems (such as "SITE" for rover or lander missions), which have multiple instances using the same name, also require REFERENCE_COORD_SYSTEM_INDEX to completely identify the reference coordinate system. For PHX, the reference is usually a SITE frame.	"LOCAL_LEVEL_FRAME", "PAYLOAD_FRAME" <u>RDRs</u> ""SITE_FRAME", "LOCAL_LEVEL_FRAME", "PAYLOAD_FRAME"	**<u>LOCATION</u>** • GEOMETRIC_CAMERA_MODEL (Group) • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group) • PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) • SITE_DERIVED_GEOMETRY_PARMS (Group) • DERIVED_IMAGE_PARMS (Group) • SURFACE_PROJECTION_PARMS (Group) • SURFACE_MODEL_PARMS (Group)	**<u>UNITS</u>** n/a
REFERENCE_COORD_SYSTEM_SOLN_ID	???		**<u>SOURCE</u>** Image Processing Software **<u>LOCATION</u>** • GEOMETRIC_CAMERA_MODEL (Group) • PAYLOAD_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group) • PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) • SITE_DERIVED_GEOMETRY_PARMS (Group) • DERIVED_IMAGE_PARMS (Group) • SURFACE_PROJECTION_PARMS (Group) • SURFACE_MODEL_PARMS (Group)	**<u>TYPE</u>** string **<u>UNITS</u>** n/a
REFERENCE_PIXEL_IMAGE_ID	Specifies the value of PRODUCT_ID for the reference pixel EDR that was used to remove bias in generating the RDR. Note: If the model rather than a reference pixel EDR was used to remove the bias then this keyword is not included in the PDS label.		**<u>SOURCE</u>** Image Processing Software **<u>LOCATION</u>** DERIVED_IMAGED_PARMS (Group)	**<u>TYPE</u>** string **<u>UNITS</u>** n/a
RELEASE_ID	Specifies the unique identifier associated with the release to the public of all or part of a data set. The release number is associated with the data set, not the mission. When a data set is released incrementally, such as every three months during a mission, the RELEASE_ID is updated		**<u>SOURCE</u>** User Parameter **<u>LOCATION</u>** IDENTIFICATION (Class)	**<u>TYPE</u>** string **<u>UNITS</u>** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	each time part of the data set is released. The first release of a data set in the mission should have a value of "0001". For example, on PHX the first release of the SSI EDR data set on PHX will have RELEASE_ID = "0001". The next SSI EDR release will have RELEASE_ID = "0002".			
RESPONSIVITY_CONSTANT	Specifies the array of radiometirc responsivity constants used in generating higher level products, such as radiometrically corrected RDRs.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGED_PARMS (Group)	**TYPE** float array **UNITS** n/a
RESPONSIVITY_CONSTANTS_FILE_NM	Specifies the name of the responsivity constants file used in generating the RDR.		**SOURCE** Image Processing Software **LOCATION** DERIVED_IMAGED_PARMS (Group)	**TYPE** string **UNITS** n/a
ROVER_MOTION_COUNTER	Specifies a set of integers which describe a (potentially) unique location (position/orientation) for a rover or lander. Each time something happens that moves, or could potentially move, the rover or lander, a new motion counter value is created. This includes intentional motion due to drive commands, as well as potential motion due to other articulating devices, such as arms or antennae. This motion counter (or part of it) is used as a reference to define instances of coordinate systems which can move such as SITE or PAYLOAD frames. The motion counter is defined in a mission-specific manner. Although the original intent was to have incrementing indices (e.g. MER), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values. For PHX, the motion counter consists of two values. In order, they are Site and Position. The Site value increments whenever a new major Site frame is declared. The Position value increments whenever ground personnel decide the lander has moved, but choose not to declare a new Site. See Section 6.3 for details on how the RMC in managed for		**SOURCE** External file (config file) **LOCATION** IDENTIFICATION (Class)	**TYPE** integer array[2] **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	PHX.			
ROVER_MOTION_COUNTER_NAME	Specifies an array that provides the formal names identifying each integer in ROVER_MOTION_COUNTER.	("SITE", "POSITION")	**SOURCE** Static Value **LOCATION** IDENTIFICATION (Class)	**TYPE** string array[2] **UNITS** n/a
SAMPLE_BITS	Specifies the stored number of bits, or units of binary information, contained in a LINE_SAMPLE value.	"1", "2", "4", "8", "16", "32", "64"	**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** integer **UNITS** n/a
SAMPLE_BIT_MASK	Specifies the active bits in a sample. Note: In the PDS, the domain of SAMPLE_BIT_MASK is dependent upon the currently-described value in the SAMPLE_BITS element and only applies to integer values. For an 8-bit sample returned as a signed 16-bit integer, only the 8 lower order bits are active, so the SAMPLE_BIT_MASK would be 2#0000000011111111#.		**SOURCE** Constant: • image_dsize **LOCATION** IMAGE (Object)	**TYPE** non-decimal **UNITS** n/a
SAMPLE_BIT_METHOD	Specifies the method in which bit scaling is performed. For PHX, the bit scaling is a 12-bit to 8-bit scaling and can be performed onboard via hardware, software or both. Note that values "HARDWARE_INVERTED" and "SOFTWARE_INVERTED" indicate that an Inverse Lookup Table (ILUT) was applied during ground processing to 8-bit data, scaling the lowest 8 bits in the signed 16-bit integer to the lowest 12 bits. This characterizes the Scaled 12-bit version of the Science EDR (SOAS), and the Inverse LUT RDR (OPGS).	"NONE", "SOFTWARE_INVERTED", "SHIFT_BY_FOUR"	**SOURCE** comp_mode **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string **UNITS** n/a
SAMPLE_BIT_MODE_ID	Specifies the type of pixel scaling performed. For PHX, pixel scaling is accomplished by using onboard lookup tables or by shifting a specified bit into the least significant bit.	"NONE", "LUT3", "LSB_BIT3"	**SOURCE** comp_lut_idx **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	0 = NONE - no scaling ; keep as 12-bit (if available) LUT3 - use lookup table 3 LSB_BIT3 – shift to make bit 3 least significant bit			
SAMPLE_CAMERA_MODEL_OFFSET	Specifies the location of the image origin with respect to the camera model's origin. For CAHV/CAHVOR models, this origin is not the center of the camera, but is the upper-left corner of the "standard"-size image, which is encoded in the CAHV vectors. (MIPL Projections - Perspective)		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixel (<pixel> unit tag required)
SAMPLE_PROJECTION_OFFSET	Specifies the sample coordinate of the location in the image of the "special" point of the mosaic. For Polar projections, this is the nadir of the polar projection. For Vertical and Orthographic projections, this is the origin of the projected coordinate system grid (i.e., X=0.0, Y=0.0). Not applicable to other projections.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixel
SAMPLE_TYPE	Specifies the data storage representation of sample value. The valid values are platform dependent. Suns and Macs and JAVA are MSB and IEEE_REAL. INTEL based machines usually running windows or linux are LSB integers and PC_REAL.	"IEEE_REAL", "MSB_INTEGER", "MSB_UNSIGNED_INTEGER", "PC_REAL", "LSB_INTEGER", "LSB_UNSIGNED_INTEGER"	**SOURCE** Calculation: • platform **LOCATION** IMAGE (Object)	**TYPE** string(30) **UNITS** n/a
SEQUENCE_ID	Specifies an identification of the spacecraft sequence associated with the given product. This keyword is for historical reasons only and should be ignored.	"N/A"	**SOURCE** static **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a
SHUTTER_CORRECTION_MODE_ID	Specifies whether shutter subtraction will be performed.	"NONE", "SHUTTER", "BIAS"	**SOURCE** Table Lookup **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string **UNITS** n/a
SHUTTER_EFFECT_CORRECTION_FLAG	Specifies whether or not a shutter effect correction was applied to the image. The shutter effect correction involves the removal from the image of the shutter, or fixed-pattern.	0 = "FALSE" 1 = "TRUE"	**SOURCE** correction **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string(5) **UNITS** n/a
SOLAR_AZIMUTH	Specifies one of two angular measurements indicating the direction to the Sun as measured from a specific point on the surface of a planet (ex., from a rover	"0.0" to "359.99" "N/A" if any SPICE kernel is unavailable.	**SOURCE** Calculated from SPICE using: • EK • SCLK	**TYPE** float **UNITS**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	or lander). The azimuth is measured positively in the clockwise direction (as viewed from above) with the meridian passing through the positive spin axis of the planet (i.e., the north pole) defining the zero reference. 0 <= SOLAR_AZIMUTH <= 360. Units are degrees.		• Leapsecond • SPK • PCK • Surface Kernel **LOCATION** SITE_DERIVED_GEOMETRY (Group)	deg (<deg> unit tag required)
SOLAR_ELEVATION	Specifies one of two angular measurements indicating the direction to the Sun as measured from a specific point on the surface of a planet (ex., from a rover or lander). The positive direction of the elevation is set by the POSITIVE_ELEVATION_DIRECTION data element. It is measured from the plane which is normal to the line passing between the surface point and the planet's center of mass, and which intersects the surface point. -90 <= SOLAR_ELEVATION <= 90. Units are degrees.	"-90.0" to "90.0" "N/A" if any SPICE kernel is unavailable.	**SOURCE** Calculated from SPICE using: • EK • SCLK • Leapsecond • SPK • PCK • Surface Kernel **LOCATION** SITE_DERIVED_GEOMETRY (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
SOLAR_LONGITUDE	Specifies the value of the angle between the body_Sun line at the time of interest and the body_Sun line at the vernal equinox. This provides a measure of season on a target body, with values of 0 to 90 degrees representing northern spring, 90 to 180 degrees representing northern summer, 180 to 270 degrees representing northern autumn and 270 to 360 degrees representing northern winter. For IRAS: the geocentric ecliptic longitude (B1950) of the Sun at the start of a scan.	NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	**SOURCE** Calculated from SPICE using: • SCLK kernel • P kernel • Landing Site kernel **LOCATION** DERIVED_IMAGE_PARMS (Group)	**TYPE** float **UNITS** deg
SOLUTION_ID	Specifies the unique identifier for the solution set to which the values in the group belong. For certain kinds of information (such as pointing correction and rover/lander localization), more than one valid set of values may exist simultaneously. Each of these sets is called a "solution" to the unknown actual value. The SOLUTION_ID is used to identify which solution is being expressed by the containing group. It is recommended that projects adopt a specific naming convention for SOLUTION_ID. The convention will assist the user in identifying a specific instance of	"TELEMETRY", etc.	**SOURCE** Image Processing Software **LOCATION** • GEOMETRIC_CAMERA_MODEL (Group) • PAYLOAD_COORDINATE_SYSTEM (Group) • LOCAL_LEVEL_COORDINATE_SYSTEM (Group) • SITE_COORDINATE_SYSTEM (Group) • PAYLOAD_DERIVED_GEOMETRY_PARMS (Group) • SITE_DERIVED_GEOMETRY_PARMS (Group)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	values used. Components making up the naming convention for SOLUTION_ID include one or more or the following: date/time, group id, institution, program, purpose, request ID, user, and version. New components may be added as needed. Note: The value of SOLUTION_ID must not be reused for new solutions – it must be globally unique. Note: A single solution can cross many data sets (e.g. for a mosaic). Therefore, you may see an SOLUTION_ID with the same value appearing in many different data sets. For PHX, the naming convention consists of several components separated by underscores: "institution_user_purpose_version". Institution or user may be omitted if not relevant. Examples: "mipl_daa_Gallery-Pan_1", "texasam_lemmon_yogi-closeup_3", "mipl_rgd_sol3nav_5", "PHX_sol4_1". The last indicates the project-approved "official" solutions for that day. The special name "telemetry" is used for values telemetered from the rover or lander.			
SOURCE_ID	Specifies an identifier for the source. For PHX, the keyword value will always be “GROUND COMMANDED” since no command information comes down in telemetry.	“GROUND COMMANDED”	**SOURCE** static **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string(30) **UNITS** n/a
SPACECRAFT_CLOCK_START_COUNT	Specifies the value of the spacecraft clock at the beginning of a time period of interest. Format is ssssssssss.mmm measured in units of seconds, with the field to the right of the decimal in milliseconds, and stored internally as a floating point number.		**SOURCE** Calculation **LOCATION** IDENTIFICATION (Class)	**TYPE** string(30) **UNITS** n/a
SPACECRAFT_CLOCK_STOP_COUNT	Specifies the value of the spacecraft clock at the end of a time period of interest. Format is ssssssssss.mmm measured in		**SOURCE** Calculation **LOCATION**	**TYPE** string(30) **UNITS**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	units of seconds, with the field to the right of the decimal in milliseconds, and stored internally as a floating point number.		IDENTIFICATION (Class)	n/a
SPACECRAFT_ID	Specifies unique integer identifier for spacecraft. See INSTRUMENT_HOST_ID.	VALUE / INSTRUMENT HOST ID "84" FM "89" EM	**SOURCE** Command line input to telemproc **LOCATION** IDENTIFICATION (Class)	**TYPE** integer **UNITS** n/a
SPICE_FILE_NAME	Specifies the names of the SPICE files used in processing the data. For Galileo, the SPICE files are used to determine navigation and lighting information.		**SOURCE** User parameter **LOCATION** TELEMETRY (Class)	**TYPE** string(180) **UNITS** n/a
STANDARD_DEVIATION	Specifies the standard deviation of the DN values in the image array.		**SOURCE** Calculation **LOCATION** IMAGE (Object)	**TYPE** float **UNITS** n/a
START_AZIMUTH	Specifies the angular distance from a fixed reference position at which an image or observation starts. Azimuth is measured in a spherical coordinate system, in a plane normal to the principal axis. Azimuth values increase according to the right hand rule relative to the positive direction of the principal axis of the spherical coordinate system. When in a SURFACE_PROJECTION group, specifies the azimuth of the left edge of the output map. Applies to CYLINDRICAL and CYLINDRICAL-PERSPECTIVE projections only.	"0" to "360"	**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)
START_TIME	Specifies the date and time of the beginning of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system format..	YYYY-MM-DDThh:mm:ss[.fff]Z NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	**SOURCE** Calculation: • SCLK kernel **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a
STOP_AZIMUTH	Specifies the angular distance from a fixed reference position at which an image or observation stops. Azimuth is measured in a spherical coordinate system, in a plane normal to the principal axis. Azimuth values increase according to the right hand rule relative to the positive direction of the principal axis of the spherical coordinate system.	"0" to "360"	**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** deg (<deg> unit tag required)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	When in a SURFACE_PROJECTION group, specifies the azimuth of the right edge of the output map. Applies to CYLINDRICAL and CYLINDRICAL-PERSPECTIVE projections only.			
STOP_TIME	Specifies the date and time of the end of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system format.	YYYY-MM-DDThh:mm:ss[.fff]Z NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	**SOURCE** Calculation: • SCLK kernel **LOCATION** IDENTIFICATION (Class)	**TYPE** string **UNITS** n/a
SUBFRAME_TYPE	Specifies the method of subframing performed on the image.	Flag / Subframe / Value 0 OFF "NONE" 1 ON "ENABLED"	**SOURCE** • subframe_x_start • subframe_y_start • subframe_x_end • subframe_y_end **LOCATION** SUBFRAME_PARMS (Group)	**TYPE** string **UNITS** n/a
SUN_FIND_FLAG	Specifies whether the sun is located in the image. For PHX, applies to SSI only.	0 = "FALSE" 1 = "TRUE"	**SOURCE** sun_find_result **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** string **UNITS** n/a
SUN_FIND_ACTIVE_FLAG	Indicates that sun finding was active. For PHX, applies to SSI only.	0 = "FALSE" 1 = "TRUE"	**SOURCE** sun_find **LOCATION** INSTRUMENT_STATE_PARMS (Group)	**TYPE** Boolean **UNITS** n/a
SURFACE_GROUND_LOCATION	Specifies any point on the surface (for SURFACE_MODEL_TYPE of "PLANE"), or the center of the sphere (for the three "SPHERE" types). This point is measured in the coordinates specified by the REFERENCE_COORD_SYSTEM_* keywords in the same group.		**SOURCE** Image Processing Software **LOCATION** SURFACE_MODEL_PARMS (Group)	**TYPE** float array[3] **UNITS** n/a
SURFACE_MODEL_FILE_NAME	Specifies the name of an XYZ or Z-component RDR used as a digital elevation model onto which the data were projected.		**SOURCE** Image Processing Software **LOCATION** SURFACE_MODEL_PARMS (Group)	**TYPE** string **UNITS** n/a
SURFACE_MODEL_TYPE	Specifies the type of surface used for the reprojection performed during the mosaicing process. "INFINITY" - refers to an infinitely distant	"PLANE", "INFINITY", "SPHERE", "SPHERE1", "SPHERE2"	**SOURCE** Image Processing Software **LOCATION** SURFACE_MODEL_PARMS (Group)	**TYPE** string **UNITS** n/a

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
	"surface" in all directions and has no parameters. "PLANE" - refers to a flat plane and require the SURFACE_NORMAL_VECTOR and SURFACE_GROUND_LOCATION keywords as parameters. "SPHERE" – refers to a spherical model where the camera is at the center of the sphere. The origin is specified by SURFACE_GROUND_LOCATION, and the radius by the first element of SURFACE_NORMAL_VECTOR. "SPHERE1" – refers to a general sphere model, whose center is defined by SURFACE_GROUND_LOCATION, and radius by the first element of SURFACE_NORMAL_VECTOR. If the camera is outside the sphere, the first intersection with the sphere is used; this makes it useful for modeling hills or rocks. "SPHERE2" – just like SPHERE1, except the second intersection with the sphere is used; this makes it useful for modeling craters.			
SURFACE_NORMAL_VECTOR	Specifies a vector normal to the surface (for SURFACE_MODEL_TYPE of "PLANE"). This vector is measured in the coordinates specified by the REFERENCE_COORD_SYSTEM_* keywords in the same group. For the "SPHERE" surface model types, the first element is used to specify the radius; the other two elements are unused. This is a misuse of this keyword's definition, which is retained for historical reasons.		**SOURCE** Image Processing Software **LOCATION** SURFACE_MODEL_PARMS (Group)	**TYPE** float array[3] **UNITS** n/a
TARGET_NAME	Specifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE.	"MARS", "CALIBRATION"	**SOURCE** Calculated by algorithm to determine if looking at the calibration target. If not, then value is "MARS". **LOCATION** IDENTIFICATION (Class)	**TYPE** string(30) **UNITS** n/a
TARGET_TYPE	Specifies the type of a named target.	"CALIBRATION", "DUST", "N/A",	**SOURCE**	**TYPE**

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
		"SUN", "PLANET"	Static Value – "PLANET" **LOCATION** IDENTIFICATION (Class)	string **UNITS** n/a
TELEMETRY_PROVIDER_ID	Specifies the provider and version of the telemetry data used in the generation of this data.	"SSW PHX_DP", "TTACS"	**SOURCE** User Parameter **LOCATION** TELEMETRY (Class)	**TYPE** string **UNITS** n/a
TELEMETRY_SOURCE_NAME	Specifies the name of the telemetry source used in creation of this data set.		**SOURCE** Input DP filename **LOCATION** TELEMETRY (Class)	**TYPE** string **UNITS** n/a
TELEMETRY_SOURCE_TYPE	Specifies the classification of the source of the telemetry used in creating this data set.	"SFDU", "DATA PRODUCT"	**SOURCE** User Parameter **LOCATION** TELEMETRY (Class)	**TYPE** string(12) **UNITS** n/a
X_AXIS_MAXIMUM	Specifies the value of the X coordinate of a Vertical or Orthographic projection at the top of the image. Note that +X is at the top of the image and +Y is at the right, so +X corresponds to North.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** meters (<m> unit tag required)
X_AXIS_MINIMUM	Specifies the value of the X coordinate of a Vertical or Orthographic projection at the bottom of the image.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** meters (<m> unit tag required)
Y_AXIS_MAXIMUM	Specifies the value of the Y coordinate of a Vertical or Orthographic projection at the right edge of the image.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** meters (<m> unit tag required)
Y_AXIS_MINIMUM	Specifies the value of the Y coordinate of a Vertical or Orthographic projection at the left edge of the image.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** meters (<m> unit tag required)

Keyword Name	Definition	Valid Values (quoted)	• Valid Value Source • Location in PDS Label	• Type • Units
ZERO_ELEVATION_LINE	Specifies the image line representing 0.0 degree elevation. Applies to Cylindrical projections.		**SOURCE** Image Processing Software **LOCATION** SURFACE_PROJECTION_PARMS (Group)	**TYPE** float **UNITS** pixels (<pixel> unit tag required)

APPENDIX B – Example PDS Label

```
PDS_VERSION_ID                    =   PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                       =   FIXED_LENGTH
RECORD_BYTES                      =   2048
FILE_RECORDS                      =   1040
LABEL_RECORDS                     =   11

/* POINTERS TO DATA OBJECTS */

^IMAGE_HEADER                     =   12
^IMAGE                            =   17

/* IDENTIFICATION DATA ELEMENTS       */

DATA_SET_ID                       =   "PHX-M-SSI-2-EDR-V1.0"
DATA_SET_NAME                     =   "PHOENIX MARS SURFACE STEREO IMAGER 2
                                      EDR VERSION 1.0"
FRAME_ID                          =   LEFT
FRAME_TYPE                        =   STEREO
GEOMETRY_PROJECTION_TYPE          =   RAW
IMAGE_ID                          =   "281632768"
IMAGE_TYPE                        =   REGULAR
INSTRUMENT_HOST_ID                =   PHX
INSTRUMENT_HOST_NAME              =   "PHOENIX"
INSTRUMENT_ID                     =   SSI
INSTRUMENT_NAME                   =   "SURFACE STEREO IMAGER LEFT"
INSTRUMENT_SERIAL_NUMBER          =   "111"
INSTRUMENT_TYPE                   =   "IMAGING CAMERA"
INSTRUMENT_VERSION_ID             =   FM
LOCAL_TRUE_SOLAR_TIME             =   "17:11:14"
MISSION_NAME                      =   PHOENIX
MISSION_PHASE_NAME                =   "PRIMARY MISSION"
OBSERVATION_ID                    =   "UNK"
OPS_TOKEN                         =   16#10C96000#
OPS_TOKEN_ACTIVITY                =   16#000010C9#
OPS_TOKEN_COMMAND                 =   16#00000000#
OPS_TOKEN_PAYLOAD                 =   16#00000006#
PLANET_DAY_NUMBER                 =   0
PRODUCER_INSTITUTION_NAME         =   "MULTIMISSION IMAGE PROCESSING SUBSYSTEM
                                      , JET PROPULSION LAB"
PRODUCT_CREATION_TIME             =   2008-11-04T22:53:16.000
PRODUCT_ID                        =   SS000EFF896228288_10C96L1M1
PRODUCT_VERSION_ID                =   "V1.0 D-22850"
RELEASE_ID                        =   "0001"
```

```
ROVER_MOTION_COUNTER              =   (0,0)
ROVER_MOTION_COUNTER_NAME         =   (SITE,POSITION)
SEQUENCE_ID                       =   "UNK"
SEQUENCE_VERSION_ID               =   "UNK"
SOLAR_LONGITUDE                   =   76.7414
SPACECRAFT_CLOCK_START_COUNT      =   "896228288.309"
SPACECRAFT_CLOCK_STOP_COUNT       =   "896228288.512"
START_TIME                        =   2008-05-26T00:17:02.333
STOP_TIME                         =   2008-05-26T00:17:02.536
TARGET_NAME                       =   MARS
TARGET_TYPE                       =   PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID            =   92
APPLICATION_PROCESS_NAME          =   APID_SSI_KEY_2
EARTH_RECEIVED_START_TIME         =   2008-05-26T02:43:53.230
EARTH_RECEIVED_STOP_TIME          =   2008-05-26T02:44:22.222
EXPECTED_PACKETS                  =   257
PACKET_MAP_MASK                   =   "NULL"
RECEIVED_PACKETS                  =   256
SPICE_FILE_NAME                   =   "phx_furnish.latest"
TELEMETRY_PROVIDER_ID             =   TTACS
TELEMETRY_SOURCE_NAME             =   "N/A"
TELEMETRY_SOURCE_TYPE             =   SFDU

/* HISTORY DATA ELEMENTS */

PROCESSING_HISTORY_TEXT           =   "CODMAC LEVEL 1 to LEVEL 2 CONVERSION
                                      VIA JPL/MIPL PHXTELEMPROC"
SOFTWARE_NAME                     =   PHXTELEMPROC
SOFTWARE_VERSION_ID               =   "V4.0 07-10-08"

/* CAMERA_MODEL DATA ELEMENTS *       /

GROUP                             =   GEOMETRIC_CAMERA_MODEL_PARMS
CALIBRATION_SOURCE_ID             =   "UNK"
MODEL_TYPE                        =   CAHVOR
MODEL_COMPONENT_ID                =   (C,A,H,V,O,R)
MODEL_COMPONENT_NAME              =   (CENTER,AXIS,HORIZONTAL,VERTICAL,
                                      OPTICAL,RADIAL)
MODEL_COMPONENT_UNIT              =   (METER,"N/A",PIXEL,PIXEL,"N/A","N/A")
MODEL_COMPONENT_1                 =   (-0.407223,0.0452166,-0.850772)
MODEL_COMPONENT_2                 =   (0.332918,0.289562,0.897396)
MODEL_COMPONENT_3                 =   (-2425.23,3454.29,384.198)
MODEL_COMPONENT_4                 =   (-2805.32,-2144.84,2336.07)
MODEL_COMPONENT_5                 =   (0.31686,0.285039,0.904629)
MODEL_COMPONENT_6                 =   (0.000323,-0.020572,-0.272812)
FILTER_NAME                       =   SSI_L1_672NM
REFERENCE_COORD_SYSTEM_INDEX      =   (0,0)
REFERENCE_COORD_SYSTEM_NAME       =   PAYLOAD_FRAME
```

SOLUTION_ID	=	InitialLanded
END_GROUP	=	GEOMETRIC_CAMERA_MODEL_PARMS
/* COORDINATE SYSTEM STATE: PAY	LO AD	*/
GROUP	=	PAYLOAD_COORDINATE_SYSTEM_PARMS
COORDINATE_SYSTEM_INDEX	=	(0,0)
COORDINATE_SYSTEM_INDEX_NAME	=	(SITE,POSITION)
COORDINATE_SYSTEM_NAME	=	PAYLOAD_FRAME
ORIGIN_OFFSET_VECTOR	=	(0.0,0.0,0.0)
ORIGIN_ROTATION_QUATERNION	=	(0.99998,-0.002264,2.1e-05,-0.005922)
POSITIVE_AZIMUTH_DIRECTION	=	CLOCKWISE
POSITIVE_ELEVATION_DIRECTION	=	UP
REFERENCE_COORD_SYSTEM_INDEX	=	0
REFERENCE_COORD_SYSTEM_NAME	=	SITE_FRAME
END_GROUP	=	PAYLOAD_COORDINATE_SYSTEM_PARMS
/* ARTICULATION DEVICE STATE: F	ILT E	R */
GROUP	=	SSI_FILTER_ARTICULATION_STATE_PARMS
ARTICULATION_DEVICE_ID	=	"SSI FILTER"
ARTICULATION_DEVICE_NAME	=	"FILTER ACTUATORS"
ARTICULATION_DEV_MOTOR_CLICKS	=	(0,180)
	E	
ARTICULATION_DEV_MTR_CLCK_NAM	=	("FILTER_POS-CURRENT", "FILTER_POS-PREVIOUS")
ARTICULATION_DEV_POSITION	=	(0,"N/A")
ARTICULATION_DEV_POSITION_ID	=	(SSI_L1_672NM,"N/A")
	E	
ARTICULATION_DEV_POSITION_NAM	=	("LEFT SSI FILTER","RIGHT SSI FILTER")
	E	
ARTICULATION_DEVICE_COMP_STAT	=	CLOCKWISE
ARTICULATION_DEVICE_COMP_NAME	=	FILTER_DIR
END_GROUP	=	SSI_FILTER_ARTICULATION_STATE_PARMS
/* ARTICULATION DEVICE STATE: S	SI *	/
GROUP	=	SSI_ARTICULATION_STATE_PARMS
ARTICULATION_DEVICE_ID	=	SSI
ARTICULATION_DEVICE_NAME	=	"SURFACE STEREO IMAGER"
ARTICULATION_DEVICE_ANGLE	=	(0.640344 <rad>,-1.10654 <rad>, 0.646107 <rad>,-1.11069 <rad>, 4.23075 <rad>,-0.60723 <rad>)
	E	
ARTICULATION_DEVICE_ANGLE_NAM	=	("AZIMUTH-BACKLASH", "ELEVATION-BACKLASH","AZIMUTH-MEASURED" ,"ELEVATION-MEASURED","AZIMUTH-INITIAL" ,"ELEVATION-INITIAL")
ARTICULATION_DEV_MOTOR_CLICKS	=	(436,-216,-100,-116)

```
                                         E
ARTICULATION_DEV_MTR_CLCK_NAM            =   ("AZIMUTH-MEASURED",
                                             "ELEVATION-MEASURED","AZIMUTH-INITIAL",
                                             "ELEVATION-INITIAL")
ARTICULATION_DEVICE_MODE                 =   DEPLOYED
                                         E
ARTICULATION_DEVICE_COMP_STAT            =   ("COUNTER-CLOCKWISE",DOWN)
ARTICULATION_DEVICE_COMP_NAME            =   (AZ_DIR,EL_DIR)
END_GROUP                                =   SSI_ARTICULATION_STATE_PARMS

/* SUBFRAME */

GROUP                                    =   SUBFRAME_PARMS
SOURCE_ID                                =   "GROUND COMMANDED"
SUBFRAME_TYPE                            =   NONE
FIRST_LINE                               =   1
FIRST_LINE_SAMPLE                        =   1
LINES                                    =   1024
LINE_SAMPLES                             =   1024
END_GROUP                                =   SUBFRAME_PARMS

/* INSTRUMENT STATE RESULTS */

GROUP                                    =   INSTRUMENT_STATE_PARMS
AZIMUTH_FOV                              =   13.8882 <deg>
ELEVATION_FOV                            =   13.9035 <deg>
DETECTOR_FIRST_LINE                      =   1
DETECTOR_LINES                           =   1024
DOWNSAMPLE_METHOD                        =   None
EXPOSURE_COUNT                           =   0
EXPOSURE_DURATION                        =   204.0 <ms>
EXPOSURE_DURATION_COUNT                  =   40
FILTER_NAME                              =   SSI_L1_672NM
FILTER_NUMBER                            =   "1"
ELECTRONICS_BIAS                         =   0
INSTRUMENT_MODE_ID                       =   "FULL_FRAME"
INSTRUMENT_TEMPERATURE                   =   (-32.5375 <degC>,-32.185 <degC>,
                                             -33.2607 <degC>,-11.0655 <degC>,
                                             -32.1496 <degC>,-52.0749 <degC>,
                                             -32.2742 <degC>)
INSTRUMENT_TEMPERATURE_COUNT             =   (-151,-144,824,-583,-660,-1334,-168)
INSTRUMENT_TEMPERATURE_NAME              =   (SSI_CCD_START,SSI_CCD_END,
                                             SSI_OPTICAL_BENCH,SSI_AZIMUTH_ACTUATOR,
                                             SSI_ELEVATION_ACTUATOR,
                                             SSI_FILTER_WHEEL_ACTUATOR,SSI_PCB_END)
OFFSET_NUMBER                            =   4065
PIXEL_AVERAGING_HEIGHT                   =   1
PIXEL_AVERAGING_WIDTH                    =   1
SAMPLE_BIT_METHOD                        =   NONE
SAMPLE_BIT_MODE_ID                       =   "N/A"
                                         G
SHUTTER_EFFECT_CORRECTION_FLA            =   FALSE
```

```
SUN_FIND_FLAG                      =   FALSE
SUN_FIND_ACTIVE_FLAG               =   FALSE
END_GROUP                          =   INSTRUMENT_STATE_PARMS

/* COMPRESSION RESULTS */

GROUP                              =   COMPRESSION_PARMS
ERROR_PIXELS                       =   0
AC_CMPRS_TABLE_ID                  =   1
DC_CMPRS_TABLE_ID                  =   1
CMPRS_QUANTZ_TBL_ID                =   0
INST_CMPRS_MODE                    =   1
INST_CMPRS_NAME                    =   "JPEG HUFFMAN ENCODING"
INST_CMPRS_QUALITY                 =   225
INST_CMPRS_RATE                    =   0.0989227
INST_CMPRS_RATIO                   =   7
INST_CMPRS_SEGMENTS                =   257
INST_CMPRS_SEG_FIRST_LINE          =   (1,0,8,8,16,16,24,24,32,32,40,40,48,48,
                                       56,56,64,64,72,72,80,80,88,88,96,96,104
                                       ,104,112,112,120,120,128,128,136,136,
                                       144,144,152,152,160,000,000,000,000,000
                                       ,184,184,192,192,200,200,208,208,216,
                                       216,224,224,232,232,000,000,000,000,000
                                       ,256,264,264,272,272,280,280,288,288,
                                       296,296,304,304,312,000,000,000,000,000
                                       ,336,336,344,344,352,352,360,360,368,
                                       368,376,376,384,384,000,000,000,000,000
                                       ,408,416,416,424,424,432,432,440,440,
                                       448,448,456,456,464,000,000,000,000,000
                                       ,488,488,496,496,504,504,512,512,520,
                                       520,528,528,536,536,000,000,000,000,000
                                       ,560,568,568,576,576,584,584,592,592,
                                       600,600,608,608,616,000,000,000,000,000
                                       ,640,640,648,648,656,656,664,664,672,
                                       672,680,680,688,688,000,000,000,000,000
                                       ,712,720,720,728,728,736,736,744,744,
                                       752,752,760,760,768,000,000,000,000,000
                                       ,792,792,800,800,808,808,816,816,824,
                                       824,832,832,840,840,000,000,000,000,000
                                       ,864,872,872,880,880,888,888,896,896,
                                       904,904,912,912,920,000,000,000,000,000
                                       ,944,944,952,952,960,960,968,968,976,
                                       9,769,849,849,929,920,000,000,000,000,000
                                       ,1016,1016)
                                   P
INST_CMPRS_SEG_FIRST_LINE_SAM      =   (1024,512,0,512,0,512,0,512,0,512,0,512
                                       ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                       512,0,512,0,512,0,512,0,512,0,512,0,512
                                       ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                       512,0,512,0,512,0,512,0,512,0,512,0,512
                                       ,0,512,0,512,0,512,0,512,0,512,0,512,0,
```

```
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512,0,512,0,
                                    512,0,512,0,512,0,512,0,512,0,512,0,512
                                    ,0,512,0,512,0,512,0,512,0,512)
INST_CMPRS_SEG_LINES                8
INST_CMPRS_SEG_SAMPLES          =   512
INST_CMPRS_SEG_MISSING_PIXELS   =   (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                    0,0,0)
INST_CMPRS_SEGMENT_QUALITY      =   4096
INST_CMPRS_SEGMENT_STATUS       =   ("0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
                                    ,"0","0","0","0","0","0","0","0","0",
                                    "0","0","0","0","0","0","0","0","0","0"
```

```
                                        ,"0","0","0","0","0","0","0","0","0",
                                        "0","0","0","0","0","0","0","0","0","0"
                                        ,"0","0","0","0","0","0","0","0","0",
                                        "0","0","0","0","0","0","0","0","0","0"
                                        ,"0","0","0","0","0","0","0","0","0",
                                        "0","0","0","0","0","0","0","0","0","0"
                                        ,"0","0","0","0","0","0","0","0","0")
END_GROUP                          =    COMPRESSION_PARMS

                                   TS
/* DERIVED GEOMETRY DATA ELEMEN     :   PAYLOAD FRAME */

GROUP                              =    PAYLOAD_DERIVED_GEOMETRY_PARMS
INSTRUMENT_AZIMUTH                 =    40.1895 <deg>
INSTRUMENT_ELEVATION               =    -63.4341 <deg>
REFERENCE_COORD_SYSTEM_INDEX       =    (0,0)
REFERENCE_COORD_SYSTEM_NAME        =    PAYLOAD_FRAME
END_GROUP                          =    PAYLOAD_DERIVED_GEOMETRY_PARMS

                                   TS
/* DERIVED GEOMETRY DATA ELEMEN     :   SITE FRAME */

GROUP                              =    SITE_DERIVED_GEOMETRY_PARMS
INSTRUMENT_AZIMUTH                 =    39.9036 <deg>
INSTRUMENT_ELEVATION               =    -63.2653 <deg>
REFERENCE_COORD_SYSTEM_INDEX       =    0
REFERENCE_COORD_SYSTEM_NAME        =    SITE_FRAME
SOLAR_AZIMUTH                      =    268.291 <deg>
SOLAR_ELEVATION                    =    27.125 <deg>
END_GROUP                          =    SITE_DERIVED_GEOMETRY_PARMS

/* IMAGE DATA ELEMENTS */

OBJECT                             =    IMAGE
INTERCHANGE_FORMAT                 =    BINARY
LINES                              =    1024
LINE_SAMPLES                       =    1024
SAMPLE_TYPE                        =    MSB_INTEGER
SAMPLE_BITS                        =    16
BANDS                              =    1
BAND_STORAGE_TYPE                  =    BAND_SEQUENTIAL
CHECKSUM                           =    1.73E+08
FIRST_LINE                         =    1
FIRST_LINE_SAMPLE                  =    1
INVALID_CONSTANT                   =    0
MAXIMUM                            =    818
MEAN                               =    165.079
MEDIAN                             =    128
MINIMUM                            =    7
MISSING_CONSTANT                   =    0
SAMPLE_BIT_MASK                    =    2#0000111111111111#
```

STANDARD_DEVIATION	=	110.204
END_OBJECT	=	IMAGE
/* IMAGE HEADER DATA ELEMENTS *	/	
OBJECT	#NAME?	AGE_HEADER
HEADER_TYPE	#NAME?	CAR2
INTERCHANGE_FORMAT	#NAME?	CII
BYTES	10	240
	=	
^DESCRIPTION	"V	ICAR2.TXT"
END_OBJECT	#NAME?	AGE_HEADER
END		