Mars Science Laboratory
Software Interface Specification
Alpha Particle X-ray Spectrometer (APXS) Experiment Data Record (EDR)

Version 2.0
JPL D-69261
MSL 576-3504

June 25, 2013

Prepared by:

_____
Helen Mortensen, OPGS

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

SIGNATURES

Prepared by:

Custodian:

Helen Mortensen, OPGS MSL Project — Date

Concurrence:

MSL Science Manager:

Joy Crisp — Date

Instrument Scientist:

Albert Yen — Date

Principal Investigator:

Ralf Gellert — Date

MSL OPGS Cognizant Engineer:

Maher Hanna — Date

PDS Program Manager:

Thomas Morgan — Date

Geosciences Node Data Manager:

Raymond Arvidson — Date

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
08/06/10	All	First draft	Draft v0.1
2/15/11	1.3, 2.3.4, 2.4.2, Fig 3, Fig 4, Appendix A &B	Updates to reference document, new filenaming convention, time standards, example label and Definitions	Draft v0.2
4/23/11	Section 2 and 3, Appendix A	Updated with comments from Albert Yen. Changed PDS label to FIXED_LENGTH with offsets.	Draft v0.3
5/24/11	Appendix A	Updated label and FMT examples.	Draft v0.4
9/1/11	Section 2	Added coordinate system	Draft v0.5
11/14/11	Appendix B	Remove MER references	Initial 1.0
Mar-2012	All	Incorporated comments from reviewers	V1.1
Apr-2012	Sec 2.4.10, fig 3, table 4, sec 4.1	Incorporated comments from Albert Yen on v1.1 release	V1.2.1
2/7/13	All	Updated missing information and updated to reflect current labels. Addresses PDS Peer review comments.	V1.3
3/26/13	Sec 1.3, 2.1	Corrected version, date, and title for applicable documents 8 and 9. Corrected statement about detection of elements with Z equal or larger than Na.	V1.4
6/25/13	Appendix A.3, B	Fixes issues identified by Peer Review	V2.0

TBD ITEMS

SECTION	DESCRIPTION
All	Identified by red text
2.3.1	Question about usage of CODMAC

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
APSS	Activity Planning and Sequencing Subsystem
APXS	Alpha Particle X-ray Spectrometer
CODMAC	Committee on Data Management and Computation
CWA	Current Working Area
EDR	Experiment Data Record
FEI	File Exchange Interface
FSW	Flight Software
ICD	Interface Control Document
ISO	International Standards Organization
JPL	Jet Propulsion Laboratory
Kbyte	Kilobytes
LSB	Least Significant Byte
MB	Mega Bytes
MIPL	Multi-mission Image Processing Laboratory
MPCS	Mission data Processing and Control Product System
MSB	Most Significant Byte
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
ODS	Operations Data Store
OPGS	Operations Product Generation Subsystem
PDS	Planetary Data System
PEL	Payload Element Lead
PPPCS	Pointing, Positioning, Phasing & Coordinate Systems
RA	Robotic Arm
RAM	Random Access Memory
RCE	Rover Compute Element
RDR	Reduced Data Record
RSVP	Rover Sequence and Visualization Program
SA-SPaH	Sample Acquisition – Sample Processing and Handling
MSLICE	Science Activity Planner
SCM	Spacecraft Configuration Manager
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SOAS	Science Operations Analysis Subsystem
SPaH	Sample Processing and Handling
TBD	To Be Determined
TC	Temperature Compensation
TDS	Telemetry Delivery Subsystem
URL	Universal Resource Locator
WEB	Warm Electronics Box

GLOSSARY

TERM	DEFINITION
Meta-Data	Selected or summary information about data. PDS catalog objects and data product labels are forms of meta-data for summarizing important aspects of data sets and data products.

1. INTRODUCTION

1.1 Purpose and Scope

The purpose of this data product Software Interface Specification (SIS) is to provide users of the Alpha Particle X-ray Spectrometer (APXS) Experiment Data Record (EDR) with a detailed description of the product and a description of how it was generated, including data sources and destinations. An APXS EDR contains data for thirteen measurements, along with engineering data. The APXS EDR data are stored in binary format. The APXS science team will produce a set of APXS Reduced Data Record (RDR) products in ASCII format. The RDRs will be described in a separate SIS document.

This SIS is intended to provide enough information to enable users to understand the APXS EDR data product. The users for whom this SIS is intended are software developers of the programs used in generating/consuming the EDR products and scientists who will analyze the data, including those associated with the Mars Science Laboratory (MSL) Project and those in the general planetary science community.

1.2 Contents

This data product SIS describes how the MSL APXS instrument acquires its data, and how the data are processed, formatted, labeled, and uniquely identified. The document discusses standards used in generating the product and software that may be used to access the product. The data product structure and organization is described in sufficient detail to enable a user to read the product. Finally, an example of a product label is provided.

1.3 Applicable Documents and Constraints

This data product SIS is responsive to the following MSL documents:

1. Mars Science Laboratory Archive Generation, Validation and Transfer Plan, Joy Crisp, and P. Theisinger, MSL-214-1333, JPL D-35281, May 28, 2010.
2. MSL Instrument Standard Electrical and Interface Specification, MSL-336-0314, JPL D-27193, April 3, 2007.
3. APXS Functional Design Document, MSL 375-1232, JPL D-34222, July 27, 2010.
4. Pointing, Positioning, Phasing & Coordinate Systems (3PCS), Volume 1, Santi Udomkesmalee, MSL-376-1297, JPL D-34642, May 29, 2007.
5. Mars Science Laboratory Surface Attitude, Positioning, and Pointing Functional Design Description (FDD), Steve Peters, MSL-376-1089, JPL D-34217, December 13, 2010.
6. Position Localization and Attitude Correction Estimate Storage (PLACES) User's Guide, JPL D-7112, MSL-586-3653, URL:https://jplwiki.jpl.nasa.gov:8443/display/places/Home
7. MSL APXS Science Team and PDS Geosciences Node Interfaces Control Document (ICD), Version 2.0, May 15, 2007.
8. Mars Science Laboratory (MSL) Project Experiment Data Record (EDR) Archive Volume Software Interface Specification (SIS), Version 1.6, JPL D-64995, March 19, 2013.

This SIS is also consistent with the following Planetary Data System documents:

9. Planetary Data System Archive Preparation Guide, Version 1.4, JPL D-31224, April 1, 2010.
10. Planetary Science Data Dictionary Document, Version 1.81, November 24, 2010.
11. Planetary Science Data Standards Reference, Version 3.8, JPL D-7669, Part 2, February 27, 2009

Finally, this SIS is meant to be consistent with the contract negotiated between the MSL Project and the APXS Principal Investigator (PI) in which experiment data records and documentation are explicitly defined as deliverable products.

1.4 Relationships with Other Interfaces

Changes to this APXS EDR SIS document affect the products, software, and/or documents listed in Table 1.

Table 1: Product and Software Interfaces to this SIS

Name	Type P=product S=software D=document	Owner
APXS EDRs	P	OPGS/MIPL
MSLEdrGen	S	MIPL
MIPL database schema	P	MIPL
APXS RDRs	P	APXS Science Team
APXS RDR SIS	D	APXS Science Team
Other APXS Programs/Products/Documents	P/S/D	APXS Science Team

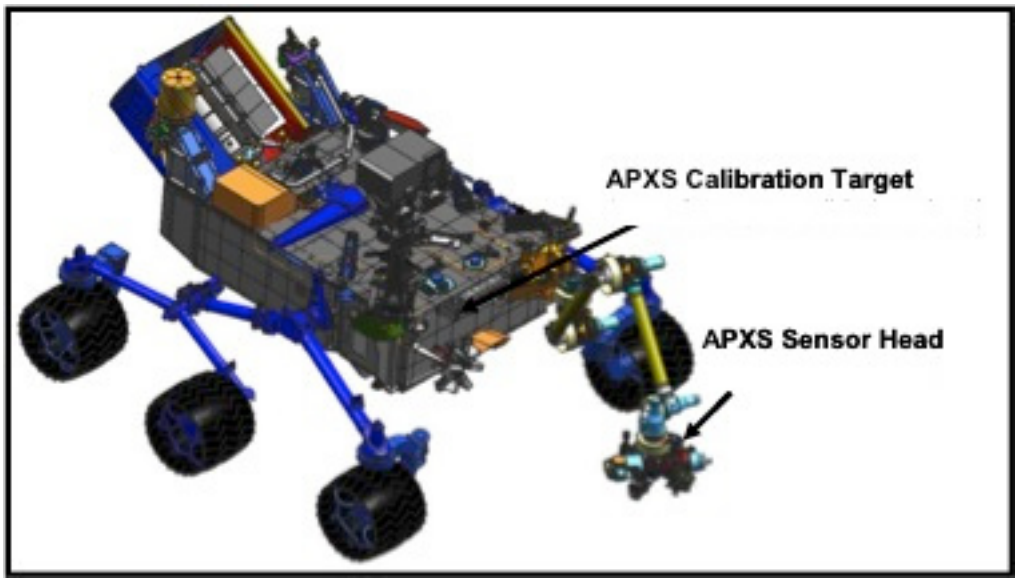
2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

2.1 Instrument Overview

The Alpha-Particle X-Ray Spectrometer (APXS) is a contact instrument that uses X-ray spectroscopy to determine the elemental composition of soils and rocks. Atmosphere argon can also be measured by the APXS. The APXS will be deployed on selected targets along the rover traverse to determine the elemental composition of samples. Materials processed by the Sample Acquisition – Sample Processing and Handling (SA/SPaH) system can also be delivered to the observation tray or to the martian surface for analysis by APXS. The APXS detects elements with Z equal or larger than Na. If present at sufficient levels, the abundances of these elements

can also be determined. Rough energy range is 800eV – 25 KeV, but it can be changed with parameter.

The instrument consists of a sensor head mounted on the rover's Robotic Arm (RA) component of the MSL SA-SPaH tools suite, and electronics mounted in the Rover Chassis. The APXS has access to onboard calibration targets for post-landing calibration. The MSL APXS includes a Peltier cooler for the detector chip that can be activated to achieve better performance at daytime ambient temperatures. A total observation time of 2 to 3 hours is necessary for a full analysis; however, it is also possible to perform a quick analysis in about 15 minutes. For those familiar with the Mars Exploration rover (MER) APXS, these MSL observations times are shorter due to the different source-detector geometry. Other significant differences from MER include the x-ray detector cooler which expands daytime operations, and the need to transfer data to the rover computer prior to instrument power down (no memory backup). Similar to operations on MER, the rover may go to sleep to conserve energy once the APXS acquisition has started; however, the rover must be awake to stop the observation and retrieve the data.



2.2 Data Product Overview

Each APXS EDR consists of a pair of files. The first file is an ASCII formatted detached PDS label, and the second file is the binary data from the instrument. Together, they are referred to as a single EDR.

The binary data file is a copy of what was in the APXS memory buffer. That is, the EDR consists of unprocessed experiment data stored in binary format along with additional Rover supplied meta-data. The instrument holds the most recent thirteen in its memory and when a new measurement begins, the oldest one is discarded. Each measurement, or spectrum, has a unique identifier. There is also a data block containing instrument’s engineering data.

There are 2 types of APXS EDRs: science, and the CWA (Current Working Area) EDRs. The science EDRs contain 13 measurements, and the engineering data. The CWA EDRs only contain the most recent measurement.

2.3 Data Processing

2.3.1 Data Processing Level

The APXS data products follow the Committee On Data Management And Computation (CODMAC) data product level numbering/convention system, to describe the processing levels/products.

APXS EDRs are considered CODMAC "Level 2" or "Edited Data" (equivalent to NASA level 0) products. EDRs are generated from "Level 1" or "Raw Data", which are the raw telemetry products . Refer to Table 2 for a breakdown of the CODMAC and NASA data processing levels.

Table 2: 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 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.

2.3.2 Data Product Generation

The APXS EDR data products will be generated by the MIPL (Multi-mission Image Processing Laboratory) at JPL under the MSL GDS/OPGS (Ground Data Systems/Operational Product Generation Sub-system) using the telemetry processing software, mslEdrGen. The EDR data products are raw un-calibrated data, reconstructed from telemetry data products in the format described by this SIS.

Meta-data collected by the rover during the data collection will be provided in PDS label. There can be multiple versions of an EDR depending on telemetry retransmissions. If telemetry data is

missing during the initial downlink from the rover memory, partial data sets may be created, and the missing data will be filled with zeroes within the EDR.

The entire EDR will be reprocessed if/when additional data are received, and new EDR, with higher version number is generated. As a rule, products with higher version numbers are, by definition, more complete.

2.3.3 Data Flow

The APXS EDR data products generated by MIPL during operations are created collectively from:

a) MPCS data products
b) SPICE kernels
c) a meta-data database.

The EDRs are stored/staged on the MSL's ODS (Operations Data Storage) which are then published into MIPL's File Exchange Interface (FEI) in real-time for secured electronic delivery and distribution to all subscribed users/sites.

After a period for data validation by the APXS team, EDRs are delivered to the PDS (NASA's Planetary Data System) for archiving and redistribution to the general public [*2*].

The size of an APXS EDR (data file) is 29818 bytes. Each EDR will be generated within 60 seconds of arrival of the raw data into the OPGS. Data will be reprocessed if some packets in the original downlink were missing and the MPCS (Multi-mission data Processing and Control System) has produced a new raw product. Partial files, created by MPCS, are filled with zeroes as necessary. The APXS EDR will be reprocessed after more data is retransmitted and regenerated by MPCS. OPGS publishes all EDRs to the FEI for distribution.

2.3.4 Labeling and Identification

The APXS EDRs are named according to the file naming convention which has been adopted by the MSL DAWG (Data Archive Working Group) for all imaging and non-imaging data products. Therefore, this naming convention applies to the APXS EDR, and several RDR data products. The file naming convention adheres to the Level II 36.3 PDS convention, which also contains a minimal level of meta-data that guarantees uniqueness, and can also be searched by various standard file directory utilities.

All MSL EDR or RDR data product can be uniquely identified by incorporating into the product filename the Rover Mission identifier, the Instrument identifier, the Starting Spacecraft Clock count (SCLK) of the observation, the data Product Type, the Site location, the rover Position within the site, the Sequence number, the camera "Eye", the spectral Filter, the product Creator identifier and a Version number. For non-camera data, several fields do not apply. The metadata fields have been selected based on MER and Phoenix mission lessons learned.

Each APXS EDR has a detached PDS label associated with the APXS binary data file. The file naming scheme for the APXS EDR and RDR data products is formed by:

<instr><config><sclk><prod><sol><site><drive><TBD><venue/who><ver>.<ext>

where,

instr = (2 alpha character) Instrument ID, denoting the source MSL science or engineering instrument that acquired the data.
Valid values for Instrument ID's are:

Valid values for:
"**AP**" - APXS

Config/ spec = (2 alphanumeric) Instrument Configuration, an operational attribute of the Instrument that assists in characterizing the data.
Valid values for MSL camera instruments:

Instrument	Configuration	
	Values	Description
APXS	"A_" "B_"	Identifies A side configuration Identifies A side configuration

sclk = (9 alphanumeric) Spacecraft Clock Start Count, in units of seconds.
Which specific SCLK is used depends on the instrument but is generally expected to be the time the data was acquired. For APXS, it is Data Product SCLK which matches the DVT (Data Validity Time) used for operational data management.

The valid values, in their progression, are as follows (non-Hex):
Range 000000000 thru 999999999 - "**000000000**", "**000000001**", **…** "**999999999**"
Range 1000000000 thru 1099999999 - "**A00000000**", "**A00000001**", **…** "**A99999999**"
Range 1100000000 thru 1199999999 - "**B00000000**", **"B00000001**", **…** **"B99999999**"
•
•
•
Range 3500000000 thru 3599999999 - "**Z00000000**", "**Z00000001**", **…** "**Z99999999**"

prod = (3 char) Product Type identifier.
This field has the following rule-of-thumb:
Beginning "**E**" - Type of **E**DR, which is the first order product with no processing applied,
The 2nd two characters identify the type of APXS EDR and weather it is complete or partial.

Valid values for Product identifiers are listed below for EDRs:

Product Type Description	Value

EDR Science and Engineering Data	"**ESC**"
EDR Current Working Area	"**ECW**"

sol = (4 numeric) Sol for the SCLK value of the EDR.

site = (3 alphanumeric) Site location count, from the RMC.

This field has the following rules-of-thumb:

a) Site - If value is any 3 alphanumeric characters, or 3 underscores (denoting value is out-of-range), then content represents Site index extracted from RMC.

The valid Site values, in their progression, are as follows (non-Hex):

Range 000 thru 999 - "**000**", "**001**", **…** "**999**"
Range 1000 thru 1099 - "**A00**", "**A01**", **…** "**A99**"
Range 1100 thru 1199 - "**B00**", **"B01**", **…** "**B99**"
•
•
•
Range 3500 thru 3599 - "**Z00**", "**Z01**", **…** "**Z99**"

drive = (4 alphanumeric) Drive (position-within-Site) location count, from the RMC.

This field has the following rules-of-thumb:

a) Drive - If value is any 4 alphanumeric characters, or 4 underscores (denoting value is out-of-range), then content represents Drive index extracted from RMC.

The valid Drive values, in their progression, are as follows (non-Hex):

Range 0000 thru 9999 - "**0000**", "**0001**", **…** "**9999**"
Range 10000 thru 10999 - "**A000**", "**A001**", **…** "**A999**"
Range 11000 thru 11999 - "**B000**", **"B001**", **… "B999**"
•
•
•
Range 35000 thru 35999 - "**Z000**", "**Z001**", **…** "**Z999**"
Range 36000 thru 36099 - "**AA00**", "**AA01**", **…** "**AA99**"
Range 36100 thru 36199 - "**AB00**", "**AB01**", **…** "**AB99**"
•
•
•
Range 38500 thru 38599 - "**AZ00**", "**AZ01**", **…** "**AZ99**"
Range 38600 thru 38699 - "**BA00**", "**BA01**", **…** "**BA99**"
Range 38700 thru 38799 - "**BB00**", "**BB01**", **…** "**BB99**"
•
•
•
Range 41100 thru 41199 - "**BZ00**", "**BZ01**", **…** "**BZ99**"
Range 41200 thru 41299 - "**CA00**", "**CA01**", **…** "**CA99**"
•
•
•

Range 65400 thru 65499 - "**LI00**", "**LI01**", **…** "**LI99**"
Range 65500 thru 65535 - "**LJ00**", "**LJ01**", **…** "**LJ35**"

_____ = (9 alphanumeric) This field is not used and remains all underscores.

venue / who = (1 alpha character) Venue and Product Producer ID shared in the same field.

Venue denotes Flight Model versus Engineering Model in data acquisition. Product Producer ID identifies the institution that generated the product.

This field has the following rules-of-thumb:

a) Venue - A value in the range "**A** - **P**" indicates Flight Model rover. A value in the range
"**Q** - **Z**" indicates Engineering (testbed) rover. The range "N - O" is not used.

b) Producer - If value is "**P**" (for Flight) or "**Y**" (for Engineering), the provider of the product is the Principal Investigator. Except for MIPL as the provider ("**M**" for Flight or "**Z**" for Engineering), the remaining characters are assigned to Co-investigator providers at the discretion of the P.I. and will be identified in due time. Within the instrument o the P.I., characters are unique. Across instruments, characters are reusable.

See the following table of valid values:

Venue		**by Producer**
Flight Model	Eng. Model	
"**M**"	"**Z**"	**M**IPL (OPGS at JPL)
"**P**"	"**Y**"	**P**rincipal Investigator of Instrument … Instrument — Principal Investigator MMM Cameras — MSSS (San Diego, CA) ChemCam RMI & SOH — LANL (Los Alamos, NM) ChemCam LIBS — CNES (France) SAM — GSFC (Goddard, MD) REMS — Ministry of Education & Science (Spain) DAN — Federal Space Agency (Russia) RAD — SwRI (Boulder, CO) CheMin — Ames Research Center (Mountain View, CA) APXS — Ralph Gellert (Canada) SA/SPaH — JPL
"**A**" - "**L**"	"**Q**" - "**X**"	Co-Investigators (to be identified by P.I. per instrument)

ver = (1 alphanumeric) Version identifier.

The valid values, in their progression, are as follows (non-Hex):

Range 1 thru 10 - "**1**", "**2**", **…** "**9**", "**0**"
Range 11 thru 36 - "**A**", "**B**", **…** "**Z**"
Range 37 and higher - "**_**" (underscore)

The Version number increments by one whenever an otherwise-identical filename would be produced. Note that not every version need exist, e.g. versions 1, 2 and 4 may exist but not 3. In general, the highest-numbered Version represents the "best" version of that product.

NOTE: To be clear, this field increments independently of all fields, including the Special
Processing field.

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

Valid values for nominal operations non-camera data products:

"**DAT**" - Non-imaging instrument data
"**LBL**" - Detached label in PDS or ODL format
"**TAB**" - **TBD** table data
-

Example #1: `APA_013760215EDR00010930008_____M1.DAT`
`APA_013760215EDR00010930008_____M1.LBL`

where,

instr	=	"**AP**"	=	APXS
config	=	"**A_**"	=	A side configuration
sclk	=	"**013760215**"	=	Spacecraft Clock Start Count of 13760215 secs
prod	=	"**EDR**"	=	Complete CWA EDR
sol	=	"0001"	=	Sol 1
site	=	"**093**"	=	Site 93
drive	=	"**008**"	=	Drive (Position-within-Site) 8
_____	=	"**_____**"	=	None
venue / who	=	"**M**"	=	Flight Model data / produced at JPL)
ver	=	"**1**"	=	Version 1
ext	=	"**DAT**"	=	Data product with PDS label

2.4 Standards Used in Generating Data Products

2.4.1 PDS Standards

The APXS EDR comply with Planetary Data System standards for file formats and labels, as specified in the PDS Standards Reference [7] and the Planetary Science Data Dictionary Document [8].

2.4.2 Time Standards

The following time standards and conventions are used throughout this document, as well as the MSL project for planning activities and identification of events.

Time Format	*Definition*

SCET	Spacecraft event time. This is the time when an event occurred on-board the spacecraft, in UTC. It is usually derived from SCLK.
SCLK	Spacecraft Clock. This is an on-board 64-bit counter, in units of nano-seconds and increments once every 100 milliseconds. Time zero corresponds to midnight on 1-Jan-1980.
ERT	Earth Received Time. This is the time when the first bit of the packet containing the current data was received at the Deep Space (DSN) station. Recorded in UCT format.
Local Solar Time	Local Solar Time (LST). This is the local solar time defined by the local solar days (sols) from the landing date using a 24 "hour" clock within the current local solar day (HR:MN:SC). Since the Mars day is 24h 37m 22s long, each unit of LST is slightly longer than the corresponding Earth unit. LST is computed using positions of the Sun and the landing site from SPICE kernels. If a landing date is unknown to the program (e.g. for calibration data acquired on Earth) then no sol number will be provided on output LST examples: SOL 12 12:00:01 SOL 132 01:22:32.498 SOL 29
RCT	Record Creation Time. This is the time when the first telemetry packet, containing a give data, set was created on the ground. Recorded in UTC format.
True Local Solar Time	This is related to LST, which is also known as the mean solar time. It is the time of day based on the position of the Sun, rather than the measure of time based on midnight to midnight "day". TLST is used in all MIPL/OPGS generated products.
SOL	Solar Day Number, also known as PLANET DAY NUMBER in PDS label. This is the number of complete solar days on Mars since landing. The landing day therefore is SOL zero.

2.4.3 Coordinate Frame Standards

The MSL Frame Manager defines several dozen coordinate frames, which can be used for commanding and pointing among other things. Refer to the Pointing, Positioning, Phasing and Coordinate Systems (PPPCS) document [4], or the Surface Attitude, Positioning and Pointing (SAPP) Functional Design Description (FDD) [5] for more details on all MSL coordinate frames. This SIS, and APXS products only use a subset of the define frames as shown in table 3.

The only place in this SIS where the full set of frames can appear is in the INSTRUMENT_COORD_FRAME_ID label, which is a command echo.

A subset of these frames needed for a specific image or data set are defined by the *_COORDINATE_SYSTEM groups.

Note that the PLACES database [6] maintains both telemetered and re-localized versions of the Site and Rover NAV frames at every available index.

Table 3 - Coordinate Frames Used for MSL Surface Operations

Frame Name (Label Keyword Value)	SHORT NAME (SAPP FDD)	REFERENCE FRAME (USED TO DEFINE)	Coordinate Frame	
			Origin	Orientation
ROVER_NAV_FRAME	RNAV	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
ROVER_MECH_FRAME	RMECH	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
LOCAL_LEVEL_FRAME	LL	Enclosing SITE_FRAME	Attached to rover (coincident with Rover Nav Frame)	North/East/Nadir
SITE_FRAME	SITE(n)	Previous SITE_FRAME	Attached to surface	North/East/Nadir
RSM_HEAD_FRAME	RSM_HEAD	ROVER_NAV_FRAME	Attached to mast head	Aligned with pointing of mast head. This corresponds to RSM_HEAD in the Frame Manager
Arm Frames: ARM_TURRET_FRAME ARM_DRILL_FRAME ARM_DRT_FRAME ARM_MAHLI_FRAME ARM_APXS_FRAME ARM_PORTION_FRAME ARM_SCOOP_TIP_FRAME ARM_SCOOP_TCP_FRAME	Arm Frames: TURRET DRILL DRT MAHLI APXS PORTION SCOOP_TIP SCOOP_TCP	ROVER_NAV_FRAME	Attached to the tool; see PPPCS for the specific tool frame.	Aligned with tool in some way; see PPPCS [Ref 1] for the specific tool Frame.

2.4.4 Rover Navigation (Rover NAV) Frame

The Rover NAV frame (RNAV) is the one used for surface navigation and mobility. By definition, the frame is attached to the rover, and moves with it when the rover moves while on the surface. Its Z origin is defined to be 0.5 mm above the deck, with the Y origin centered on the rover and the X origin aligned with the middle wheels' rotation axis for the deployed rover and suspension system on a flat plane. The +X axis points to the front of the rover, +Y to the right side, and +Z down (perpendicular to the chassis deck). See Figure 2.1.

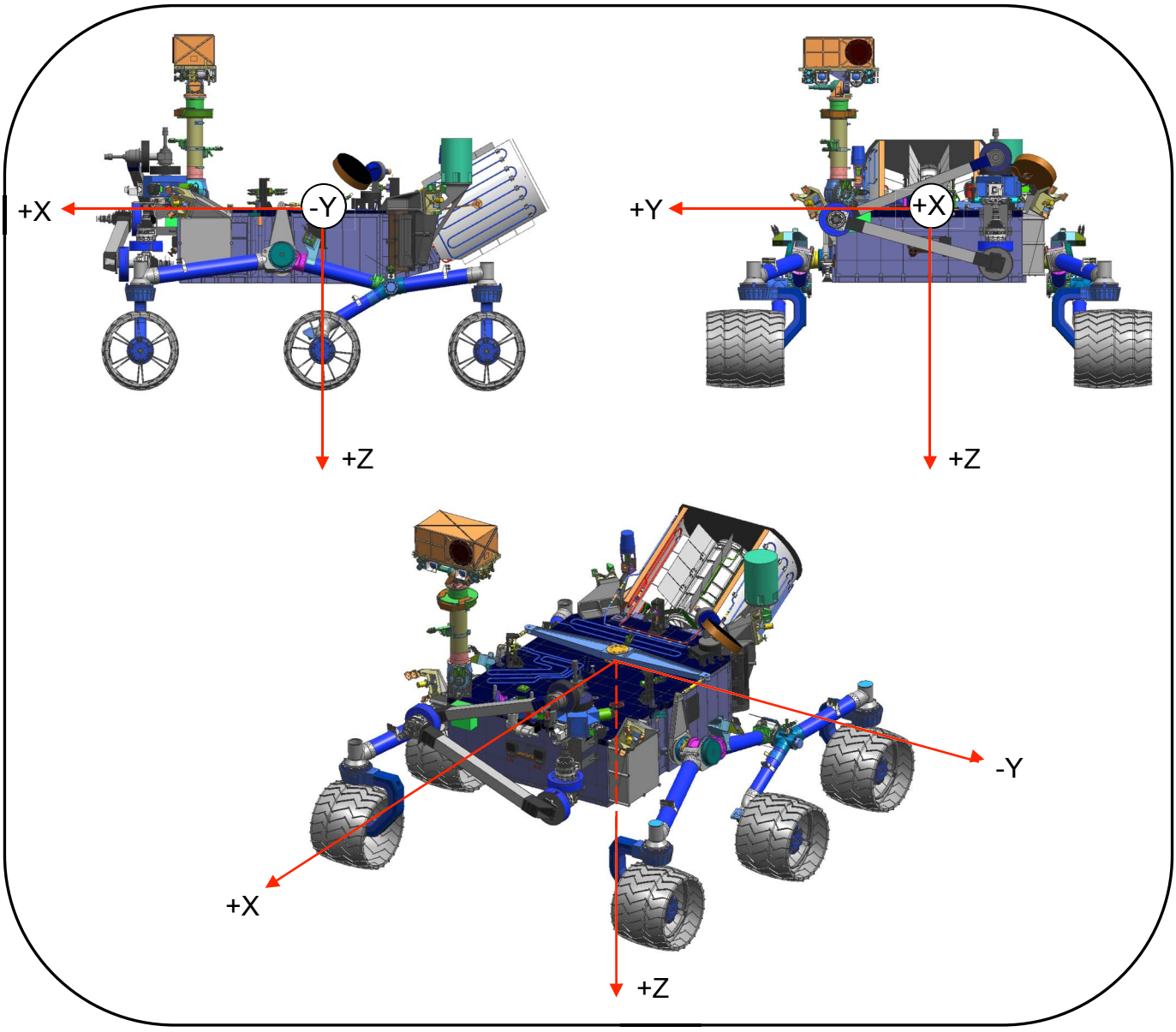


Figure 2.1: Rover Navigation (RNAV) Coordinate Frame.

The Rover NAV frame is specified via an offset from the current Site frame, and a quaternion that represents the rotation between the two. A new instance of the Rover NAV frame, with a potentially unique offset/quaternion, is created every time the ROVER_MOTION_COUNTER increments.

Orientation of the rover (and thus Rover NAV) with respect to Local Level or Site is also sometimes described by Euler angles as shown in Figure 2.2, where ψ is heading, θ is attitude or pitch, and ϕ is bank or roll.

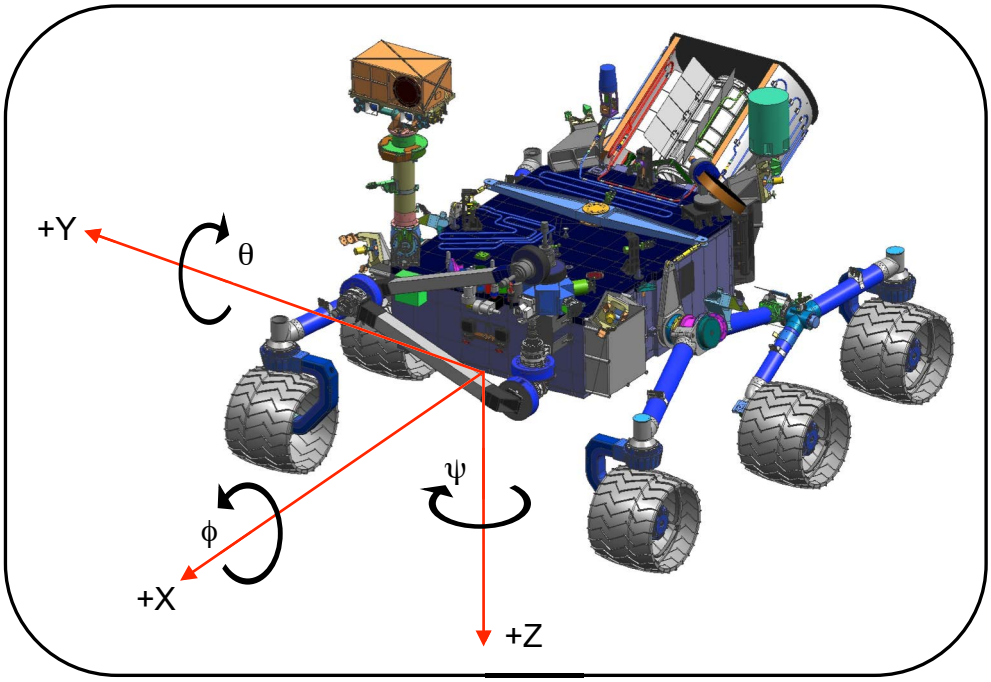


Figure 2.2: Yaw, Pitch and Roll Definitions.

2.4.5 Rover Mechanical (Rover Mech) Frame

The Rover Mechanical (RMECH) frame is oriented identically to the Rover Nav frame. The origin is forward of Rover NAV by x=0.09002 meters. In other words, given a point expressed in Rover Mech, if you add (0.09002, 0.0, 0.0) you will get the same point expressed in Rover NAV. Rover Mech is not used by any nominal products (EDR or RDR) but could appear in certain special products, generally having to do with arm kinematics.

2.4.6 Local Level Frame

The Local Level frame is coincident with the Rover Nav frame, i.e. they share the same origin at all times. The orientation is different, however. The +X axis points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system. Thus the orientation matches the orientation of Site frames.

Local Level frames are defined by an offset from the current Site frame, with an identity quaternion.

2.4.7 Site Frame

Site frames are used to reduce accumulation of rover localization error. They are used to provide a common reference point for all operations within a local area. Rover Nav

and Local Level frames are specified using an offset from this origin. When a new Site is declared, that becomes the new reference, and the offset is zeroed. In this way, long-term localization error is relegated to the offset between Sites, becoming irrelevant to local operations, because the positions are reset with each new Site.

When a Site frame is declared, it is identical to the Local Level frame, sharing both orientation and position. However, the Site frame is fixed to the Mars surface; when the rover moves, Local Level moves with it but Site stays put. Therefore, for the Site frame, +X points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system.

Sites are indexed, meaning there are multiple instances. Site 1 by definition represents the landing location. New Sites are declared as needed during operations, as the rover moves away from the local area. See Figure 2.3.

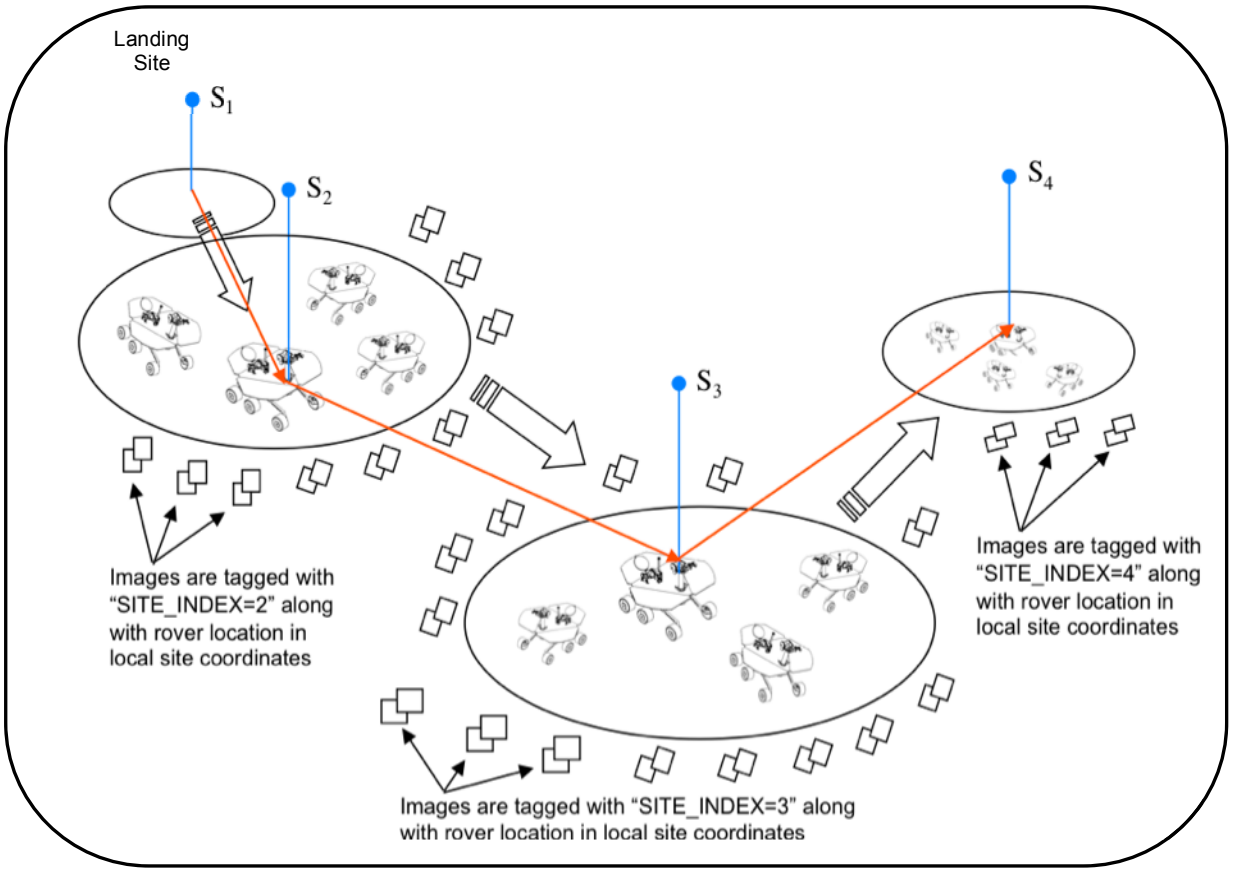


Figure 2.3: Site and Rover Frames.

The PLACES database [6] stores the set of all site-to-site offsets; such offsets are not in the image label.

2.4.8 RSM Frame

The RSM frame is attached to the Remote Sensing Mast (RSM) camera head, and moves with it. See the PPPCS for specific definition. It is expressed as an offset and quaternion from the Rover NAV frame.

2.4.9 Arm Frames

The various tool frames are attached to and aligned with the tool in some manner specific to that tool. See the PPPCS [4] for actual frame definitions.

2.4.10 Data Storage Conventions

The APXS EDR data files contain binary data. Spectral data are 16-bit unsigned integers stored in LSB first order. Temperature data are 8-bit integers. The detached PDS labels for APXS EDR's are stored as ASCII text.

2.5 Data Validation

Validation of the MSL 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. Validations 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. OPGS will perform the automatic validation.

Manual validation of the data will be performed by the science operations team, with both spot-checking of data throughout the life of the mission, and comprehensive validation of a subset of the data (for example, a couple of days' worth of data). Data products will be viewed by a human, and validation in this case will include inspection of data for completeness, consistency of label parameters, verification of meta-data values, and a general check for any problems that might not have been anticipated in the automated validation procedure.

3. DETAILED DATA PRODUCT SPECIFICATIONS

3.1 Data Product Structure and Organization

The structure of the APXS EDR consists of a detached ASCII PDS label and a binary data file as shown in Figure 2.

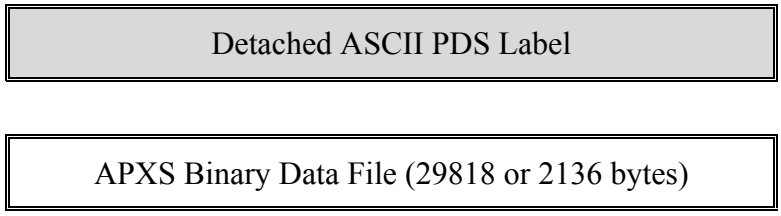


Figure 2: The APXS EDR consists of two files.

3.2 Data Format Descriptions

An APXS Science EDR data file consists of 29818 (Science and Engineering) or 2136 (CWA) bytes of APXS binary data with a detached ASCII PDS label (see sec. 3.2.1). APXS data is a copy of the instrument's memory and contains data for one (CWA) or thirteen (Science) measurements. Each measurement consists of APXS spectra and instrument state information. The APXS Science EDR also contains a 2498-byte block of engineering data from the instrument's internal memory (Figure 3).

An APXS measurement fills the 13 internal buffers as prescribed by the commanded cycle time. The cycle time can be selected to collect data for durations in excess of 18-hours, but 2 to 3 hour integration times will be more typical for MSL operations.

Science Header (42 bytes) • Ancillary Data (16 bytes) • GroupId (2 bytes) • Command Reply Frame (12 Bytes) • Science Frame Header (12 bytes)
APXS Measurement 1 (2098 bytes)
APXS Measurement 2 (2098 bytes)
. . .
APXS Measurement 12 (2098 bytes)
APXS Measurement 13 (CWA) (2098 bytes)
APXS Engineering Data (2498 bytes)
Trailing checksum(4 bytes)

Figure 3: Schematic of an APXS Science EDR data file. (29818 bytes total)

An APXS CWA (Current Working Area) EDR data file contains 2136 bytes (2098 bytes APXS CWA binary data and ancillary data) with a detached ASCII PDS label (see sec. 3.2.1). APXS data is a copy of the instrument's CWA memory and contains data for one measurement. The measurement consists of APXS spectra and instrument state information. (Figure 4).

Science Header (34 bytes) • Ancillary Data (8 bytes) • GroupId (2 bytes) • Command Reply Frame (12 Bytes) • Science Frame Header (12 bytes)
APXS Measurement (CWA) (2098 bytes)
Trailing checksum (4 bytes)

Figure 4: Schematic of an APXS CWA only EDR data file.

The detailed measurement record structure is shown in Table 4. Table 5 describes the detailed engineering data.

Table 4: MSL APXS Measurement Data Components

Bytes	Description
1-4	Spectrum sum counter. 32-bit MSB first unsigned integer.
5-8	Spectrum absolute start time in SCLK seconds with respect to the last time sync or clock second counter in case no time sync was received). 32-bit MSB first unsigned integer.
9-12	Spectrum absolute stop time in SCLK seconds with respect to the spectrum absolute start time. 32-bit MSB first unsigned integer.
13-16	Summation of the main electronics temperatures (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 228.72) – 165
17-20	Sum of main sensor head temperatures (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 190.3) – 250
21-22	Total number of main electronics temperature data/measurements. 16-bit MSB first unsigned integer.
23-24	Total number of sensor head temperature data/measurements. 16-bit MSB first unsigned integer.
25-28	Minimum main electronics temperature measured during acquisition (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 228.72) - 165

Bytes	Description
29-32	Maximum main electronics temperature measured during acquisition (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 228.72) - 165
33-36	Minimum sensor head temperature measured during acquisition (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 190.3) – 250
37-40	Maximum sensor head temperature measured during acquisition (raw ADC value). 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 190.3) – 250
41-44	High voltage rail measured at end of spectrum (raw ADC value). 32-bit MSB first unsigned integer. Voltage = - (HEX2DEC(value) / 65535) * 182.995
45-48	Back voltage bias measured at end of spectrum (raw ADC value). 32-bit MSB first unsigned integer. Voltage = - (HEX2DEC(value) / 65535) * 182.995
49-2096	Spectrum data. 16-bit MSB first unsigned integer.
2097-2098	Dead time counter in 1/10 seconds. 16-bit MSB first unsigned integer.

Table 5: MSL APXS Engineering Data Component

Bytes	Description
1- 4	Most recent time sync received or zero filled in case no time sync has been received. 32-bit MSB first unsigned integer.
5 – 20	Reserved (16 bytes)
21 – 24	Number of valid commands since power up, 32-bit MSB first unsigned integer.
25 - 28	Number of rejected commands since power up, 32-bit MSB first unsigned integer.
29 – 32	Flight software version. (0 = Gold 1, 1 = Gold 2, 3 = Silver 1, 4 = Silver 2), 32-bit MSB first unsigned integer.
33 – 36	Flight software checksum, 32-bit MSB first unsigned integer.
37 – 40	Data accumulation cycle time span in seconds, 32-bit MSB first unsigned integer.

Bytes	Description
41 – 2420	Temperature lookup table with 1190 elements, 2 bytes each, 16-bit MSB first unsigned integer.
2421 - 2424	Temperature threshold for the cooler to power on in automatic mode in degrees Celsius, 32-bit MSB first unsigned integer.
2425 – 2428	Temperature threshold for the cooler to power off in automatic mode in degress Celsius, 32-bit MSB first unsigned integer.
2429 – 2430	Comparator threshold voltage control value, 16-bit MSB first unsigned integer.
2431	High voltage generator logic used. A value of "1" denotes positive logic. A value of "0" denotes negative logic. 16-bit MSB first unsigned integer.
2432	Main Electronics Temperature High Limit, unsigned integer.
2433	Main Electronics Temperature Low Limit, unsigned integer.
2434	Sensor Head Temperature High Limit, unsigned integer.
2435	High Voltage Rail High Limit, unsigned integer.
2436	High Voltage Rail Low Limit, unsigned integer.
2437	Cable Resistance High Limit, unsigned integer.
2438	Cable Resistance Low Limit, unsigned integer.
2439	Comparator Threshold High Limit, unsigned integer.
2440	Comparator Threshold Low Limit, unsigned integer.
2441	Back Voltage High Limit, unsigned integer.
2442	Back Voltage Low Limit, unsigned integer.
2443	Cooler Voltage High Limit, unsigned integer.
2444	Cooler Voltage Low Limit, unsigned integer.

Bytes	Description
2445 – 2452	Reserved spares. (8 bytes)
2453 – 2456	Parameter Checksum using the Fletchers checksum algorithm, 32-bit MSB first unsigned integer.
2457 – 2460	Main Electronics Unit Temperature (RAW ADC value) , 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 228.72) - 165
2461 – 2464	Sensor Head (SH) Temperature (RAW ADC value) , 32-bit MSB first unsigned integer. Deg C units = (HEX2DEC(value) / 190.3) – 250
2465 – 2468	High Voltage Rail (RAW ADC value) , 32-bit MSB first unsigned integer. Voltage units = - (HEX2DEC(value) / 65535) * 182.995
2469 – 2472	Cable Resistance (RAW ADC value) , 32-bit MSB first unsigned integer. mOhm units = (HEX2DEC(value) / .3408)
2473 – 2476	Comparator Threshold Voltage (RAW ADC value) , 32-bit MSB first unsigned integer. Voltage units = (HEX2DEC(value) / 32.764)
2477 – 2480	Back Voltage Bias (RAW ADC value) , 32-bit MSB first unsigned integer. Voltage units = - (HEX2DEC(value) / 65535) * 182.995
2481 – 2484	Cooler Power Voltage (RAW ADC value) , 32-bit MSB first unsigned integer. Voltage units = (HEX2DEC(value) / 16383.75
2485 – 2488	(Redundant) software checksum, 32-bit MSB first unsigned integer
2489	Main Electronics Unit Temperature out of range test result, unsigned integer. (0 = pass, 1 = fail)
2490	Sensor Head (SH) Temperature out of range test result, unsigned integer. (0 = pass, 1 = fail)
2491	High Voltage Rail out of range test result, unsigned integer. (0 = pass, 1 = out-of-range)
2492	Cable Resistance out of range test result, unsigned integer. (0 = pass, 1 = out-of-range)
2493	Comparator Threshold Voltage out of range test result, unsigned integer. (0 = pass, 1 = out-of-range)
2494	Back Voltage Bias out of range test result, unsigned integer. (0 = pass, 1 = out-of-range)
2495	Cooler Power Voltage out of range test result, unsigned integer. (0 = pass, 1 = out-of-range)

Bytes	Description
2496	Spare Test, unsigned integer
2497	POST RAM Check result, unsigned integer. . (0 = pass, 1 = fail)
2498	POST EEPROM Check result, unsigned integer. . (0 = pass, 1 = fail)

3.3 Label and Header Descriptions

3.3.1 PDS Label

APXS EDR data products have detached PDS labels stored as ASCII. A PDS label is object-oriented and describes the objects in the data file. The PDS label contains keywords for product identification and for data object definitions. The label also contains descriptive information needed to interpret or process the data objects in the file.

PDS labels are written in Object Description Language (ODL) [7]. 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 starting record number for the data object within the file.

Each PDS keyword defined for the APXS label will always be included in the PDS label. If a keyword does not have a value, a value of NA will be given as the keyword value.

3.3.2 PDS Data Objects

An APXS EDR consists of two data objects that are described in the PDS label as tables. The first data object is a table of the APXS spectra and associated data. It has 13 rows for Science EDRs and only 1 row for CWA EDRs with each row containing an APXS measurement (Figure 3). The length of each row is 2098 bytes. The second data object in the APXS Science EDR data file is a table containing the engineering bytes from the APXS instrument. This table has one row that is 2498 bytes long.

4. APPLICABLE SOFTWARE

4.1 Utility Programs

A reader program, called apxs2asc, will be available for a variety of computer platforms to convert the APXS EDR binary format data into ASCII format. This reader will be archived with the EDR data products. The ASCII format data can be imported into spread sheet and plotting programs.

In case the program in not available, or no longer executes, the algorithm is simply to read the binary data and parse as described in tables 4 and 5.

APPENDIX A - EXAMPLE OF AN APXS SCIENCE EDR LABEL

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 29818
FILE_RECORDS                    = 1

/* POINTERS TO DATA OBJECTS */

^SCI_HEADER_TABLE               = ("APA_397764725ESC00030020000_______M1.DAT"
                                   ,1)
^SCIENCE_TABLE                  = ("APA_397764725ESC00030020000_______M1.DAT"
                                   ,43<BYTES>)
^ENGINEERING_TABLE              = ("APA_397764725ESC00030020000_______M1.DAT"
                                   ,27317<BYTES>)
^ERROR_CONTROL_TABLE            = ("APA_397764725ESC00030020000_______M1.DAT"
                                   ,29815<BYTES>)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-APXS-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS ALPHA PARTICLE
                                  X-RAY SPECTROMETER 2 EDR V1.0"
COMMAND_SEQUENCE_NUMBER         = 0
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = APXS
INSTRUMENT_NAME                 = "ALPHA PARTICLE X-RAY
                                  SPECTROMETER"
INSTRUMENT_TYPE                 = SPECTROMETER
MSL:LOCAL_MEAN_SOLAR_TIME       = "Sol-00003M14:02:23:096"
LOCAL_TRUE_SOLAR_TIME           = "14:36:00"
MISSION_NAME                    = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME               = "PRIMARY SURFACE MISSION"
OBSERVATION_ID                  = UNK
PLANET_DAY_NUMBER               = 3
PRODUCER_INSTITUTION_NAME       = "MULTIMISSION INSTRUMENT PROCESSING LAB,
                                  JET PROPULSION LAB"
PRODUCT_CREATION_TIME           = 2013-04-11T03:31:49.000
PRODUCT_ID                      = "APA_397764725ESC00030020000_______M1"
PRODUCT_VERSION_ID              = "V2.0"
PRODUCT_TYPE                    = APXS_EDR
RELEASE_ID                      = "0002"
MSL:REQUEST_ID                  = "0"
ROVER_MOTION_COUNTER            = (2,0,0,0,0,0,346,186,0,0)
ROVER_MOTION_COUNTER_NAME       = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                  DRILL, RSM, HGA, DRT, IC)
SEQUENCE_ID                     = "aut_04096"
SEQUENCE_VERSION_ID             = "0"
MSL:ACTIVE_FLIGHT_STRING_ID     = A
SOLAR_LONGITUDE                 = 152.265
SPACECRAFT_CLOCK_CNT_PARTITION  = 1
SPACECRAFT_CLOCK_START_COUNT    = "0397764256.034"
SPACECRAFT_CLOCK_STOP_COUNT     = "0397764725.040"
START_TIME                      = 2012-08-09T06:06:30.008
STOP_TIME                       = 2012-08-09T08:58:15.299
TARGET_NAME                     = MARS
TARGET_TYPE                     = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID          = 97
APPLICATION_PROCESS_NAME        = "ApxsScienceAndEng"
MSL:VIRTUAL_CHANNEL_ID          = "0"
MSL:COMMUNICATION_SESSION_ID    = "40032"
DOWNLOAD_PRIORITY               = 28
EARTH_RECEIVED_START_TIME       = 2012-08-09T08:58:15.299
MSL:EXPECTED_TRANSMISSION_PATH  = "3851"
```

```
MSL:AUTO_DELETE_FLAG             = "FALSE"
MSL:TRANSMISSION_PATH            = "65535"
FLIGHT_SOFTWARE_VERSION_ID       = "135816864"
MSL:FLIGHT_SOFTWARE_MODE         = "8"
MSL:PRODUCT_COMPLETION_STATUS    = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                  = "0"
MSL:SEQUENCE_EXECUTION_COUNT     = 0
RECEIVED_PACKETS                 = 4
SPICE_FILE_NAME                  = "chronos.msl"
TELEMETRY_PROVIDER_ID            = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM    = 57250
MSL:TELEMETRY_SOURCE_SIZE        = 29818
TELEMETRY_SOURCE_NAME           = {"ApxsScienceAndEng_0397764725-40263-1.dat"
                                  ,"ApxsStart_0397764256-34405-1.dat"}
MSL:TELEMETRY_SOURCE_TYPE        = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME   = mslsmsampcs1
EXPECTED_PACKETS                 = 4

/* COORDINATE SYSTEM STATE: ROVER AT THE START */

GROUP                            = START_ROVER_COORDINATE_SYSTEM_PARMS
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (2,0,0,0,0,0,346,176,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA, DRT, IC)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (0.0142017,-0.0352224,0.8257101,0.5628148)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX    = 2
END_GROUP                        = START_ROVER_COORDINATE_SYSTEM_PARMS

/* ARTICULATION DEVICE STATE: ROBOTIC ARM AT THE START */

GROUP                            = START_ARM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (1.0e+30<rad>, 1.0e+30<rad>,
                                    1.0e+30<rad>, 1.0e+30<rad>,
                                    1.0e+30<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                   "JOINT 2 SHOULDER ELEVATION",
                                   "JOINT 3 ELBOW-ENCODER",
                                   "JOINT 4 WRIST-ENCODER",
                                   "JOINT 5 TURRET-ENCODER")
END_GROUP                        = START_ARM_ARTICULATION_STATE_PARMS

/* COORDINATE SYSTEM STATE: ROVER AT THE END */

GROUP                            = ROVER_COORDINATE_SYSTEM_PARMS
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (2,0,0,0,0,0,346,186,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA, DRT, IC)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (0.0142017,-0.0352224,0.8257101,0.5628148)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX    = 2
END_GROUP                        = ROVER_COORDINATE_SYSTEM_PARMS

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */

GROUP                            = RSM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID          = RSM
 ARTICULATION_DEVICE_NAME        = "REMOTE SENSING MAST"
 ARTICULATION_DEVICE_ANGLE       = (3.1589742<rad>, 0.8028175<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = (AZIMUTH, ELEVATION)
END_GROUP                        = RSM_ARTICULATION_STATE_PARMS

/* ARTICULATION DEVICE STATE: ROBOTIC ARM AT THE END (DP CREATION) */
```

```
GROUP                           = ARM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID         = ARM
 ARTICULATION_DEVICE_NAME       = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE      = (1.0e+30<rad>, 1.0e+30<rad>,
                                   1.0e+30<rad>, 1.0e+30<rad>,
                                   1.0e+30<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME = ("JOINT 1 SHOULDER AZIMUTH",
                                   "JOINT 2 SHOULDER ELEVATION",
                                   "JOINT 3 ELBOW-ENCODER",
                                   "JOINT 4 WRIST-ENCODER",
                                   "JOINT 5 TURRET-ENCODER")
 CONTACT_SENSOR_STATE           = OPEN
 CONTACT_SENSOR_STATE_NAME      = "APXS SWITCH"
END_GROUP                       = ARM_ARTICULATION_STATE_PARMS


/* DATA OBJECT DEFINITION DESCRIPTIONS */

/* APXS SCIENCE HEADER OBJECT */

OBJECT                          = SCI_HEADER_TABLE
   NAME                         = SCIENCE_HEADER
   INTERCHANGE_FORMAT           = BINARY
   ROWS                         = 1
   COLUMNS                      = 11
   ROW_BYTES                    = 42
   DESCRIPTION                  = "
    This table contains science associated instrument
    data as observed by the Mars Science Laboratory (MSL)
    Alpha Particle X-ray Spectrometer (APXS).

    The complete column definitions are contained in the following external
    file found in the LABEL directory of the archive volume.
    "
   ^STRUCTURE                   = "APXS_EDR_SCI_HEADER.FMT"
END_OBJECT                      = SCI_HEADER_TABLE

/* APXS SCIENCE OBJECT */

OBJECT                          = SCIENCE_TABLE
   NAME                         = APXS_SPECTRA
   INTERCHANGE_FORMAT           = BINARY
   ROWS                         = 13
   COLUMNS                      = 15
   ROW_BYTES                    = 2098
   DESCRIPTION                  = "
    This table contains science associated instrument
    data as observed by the Mars Science Laboratory (MSL)
    Alpha Particle X-ray Spectrometer (APXS).

    The complete column definitions are contained in the following external
    file found in the LABEL directory of the archive volume.
    "
   ^STRUCTURE                   = "APXS_EDR_SCIENCE.FMT"
END_OBJECT                      = SCIENCE_TABLE

/* APXS ENGINEERING OBJECT */

OBJECT                          = ENGINEERING_TABLE
   NAME                         = APXS_ENGINEERING_DATA
   INTERCHANGE_FORMAT           = BINARY
   ROWS                         = 1
   COLUMNS                      = 45
   ROW_BYTES                    = 2498
   DESCRIPTION                  = "
    This table contains science associated instrument
    data as observed by the Mars Science Laboratory (MSL)
    Alpha Particle X-ray Spectrometer (APXS).

    The complete column definitions are contained in the following external
    file found in the LABEL directory of the archive volume.
    "
```

```
   ^STRUCTURE                    = "APXS_EDR_ENGINEERING.FMT"
END_OBJECT                       = ENGINEERING_TABLE

/* ERROR CONTROL VALUE OBJECT */

OBJECT                           = ERROR_CONTROL_TABLE
   NAME                          = ERROR_CONTROL
   INTERCHANGE_FORMAT            = BINARY
   ROWS                          = 1
   ROW_BYTES                     = 4
   COLUMNS                       = 1

     OBJECT                      = COLUMN
       NAME                      = ERROR_CONTROL_VALUE
       DATA_TYPE                 = MSB_UNSIGNED_INTEGER
       START_BYTE                = 1
       BYTES                     = 4
       DESCRIPTION               = "Contains a CRC or Fletcher checksum value
                                   as indicated in the CHMN_HSKN_HEADER_TABLE
                                   control and status error control type field
                                   ."
     END_OBJECT                  = COLUMN
END_OBJECT                       = ERROR_CONTROL_TABLE
END
```

A.1 CWA Data Objects

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 2136
FILE_RECORDS                    = 1

/* POINTERS TO DATA OBJECTS */

^CWA_HEADER_TABLE               = ("APA_397671636ECW00020010008_______M1.DAT",1)
^SCIENCE_TABLE                  = ("APA_397671636ECW00020010008_______M1.DAT",35<BYTES>)
^ERROR_CONTROL_TABLE            = ("APA_397671636ECW00020010008_______M1.DAT",2133<BYTES>)


/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-APXS-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS ALPHA PARTICLE X-RAY SPECTROMETER"
COMMAND_SEQUENCE_NUMBER         = 0
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = APXS
INSTRUMENT_NAME                 = "ALPHA PARTICLE X-RAY SPECTROMETER"
INSTRUMENT_TYPE                 = SPECTROMETER
MSL:LOCAL_MEAN_SOLAR_TIME           = "Sol-00002M12:52:27:054"
LOCAL_TRUE_SOLAR_TIME           = "13:25:49"
MISSION_NAME                    = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME              = ORT7B
OBSERVATION_ID                  = UNK
PLANET_DAY_NUMBER               = 2
PRODUCER_INSTITUTION_NAME       = "MULTIMISSION IMAGE PROCESSING LAB,
                                  JET PROPULSION LAB"
PRODUCT_CREATION_TIME           = 2012-061T16:50:39.000
PRODUCT_ID                      = "APA_397671636ESC00020010008_______M1"
PRODUCT_VERSION_ID              = "V1.0"
PRODUCT_TYPE                    = APXS_EDR
RELEASE_ID                      = "0001"
MSL:REQUEST_ID                  = 00001
ROVER_MOTION_COUNTER            = (1,8,12,0,0,0,114,26,0,0)
ROVER_MOTION_COUNTER_NAME       = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                  DRILL, RSM, HGA, DRT, IC)
SEQUENCE_ID                     = "aut_04096"
SEQUENCE_VERSION_ID             = "0"
MSL:ACTIVE_FLIGHT_STRING_ID     = A
SOLAR_LONGITUDE                 = 151.71
SPACECRAFT_CLOCK_CNT_PARTITION  = 1
SPACECRAFT_CLOCK_START_COUNT    = "0397671167.062"
SPACECRAFT_CLOCK_STOP_COUNT     = "0397671636.062"
START_TIME                      = 2012-061T13:41:43.422
STOP_TIME                       = 2012-061T13:41:43.422
TARGET_NAME                     = MARS
TARGET_TYPE                     = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID          = 98
APPLICATION_PROCESS_NAME        = "ApxsCwa"
MSL:VIRTUAL_CHANNEL_ID          = “32”
MSL:COMMUNICATION_SESSION_ID    = "30022"
DOWNLOAD_PRIORITY               = 28
EARTH_RECEIVED_START_TIME       = 2012-061T13:45:10.364
MSL:EXPECTED_TRANSMISSION_PATH  = "3851"
MSL:AUTO_DELETE_FLAG            = FALSE
MSL:TRANSMISSION_PATH           = 65535
FLIGHT_SOFTWARE_VERSION_ID      = "127216820"
MSL:FLIGHT_SOFTWARE_MODE        = "8"
MSL:PRODUCT_COMPLETION_STATUS   = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                 = "0"
MSL:SEQUENCE_EXECUTION_COUNT    = 0
RECEIVED_PACKETS                = 4
```

```
SPICE_FILE_NAME                  = "chronos.msl"
TELEMETRY_PROVIDER_ID            = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM    = 13813
MSL:TELEMETRY_SOURCE_SIZE        = 29818
TELEMETRY_SOURCE_NAME            = "ApxsCwa_0397671636-62395-1.dat"
MSL:TELEMETRY_SOURCE_TYPE        = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME   = mslsmsampcs1
EXPECTED_PACKETS                 = 4

/* COORDINATE SYSTEM STATE: ROVER AT THE START */

GROUP                            = START_ROVER_COORDINATE_SYSTEM_PARMS
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (1,8,12,0,0,0,114,26,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA, DRT, IC)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (-0.02696,0.0229748,-0.690014,0.722929)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX    = 1
END_GROUP                        = START_ROVER_COORDINATE_SYSTEM_PARMS

/* ARTICULATION DEVICE STATE: ROBOTIC ARM AT THE START */

GROUP                            = START_ARM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (1e+30<rad>, 1e+30<rad>, 1e+30<rad>, 1e+30<rad>, 1e+30<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                   "JOINT 2 SHOULDER ELEVATION",
                                   "JOINT 3 ELBOW-ENCODER",
                                   "JOINT 4 WRIST-ENCODER",
                                   "JOINT 5 TURRET-ENCODER")
END_GROUP                        = START_ARM_ARTICULATION_STATE_PARMS

/* COORDINATE SYSTEM STATE: ROVER AT THE END */

GROUP                            = ROVER_COORDINATE_SYSTEM_PARMS
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (1,8,12,0,0,0,114,26,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA, DRI, IC)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (-0.02696,0.0229748,-0.690014,0.722929)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX    = 1
END_GROUP                        = ROVER_COORDINATE_SYSTEM_PARMS

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */

GROUP                            = RSM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID          = RSM
 ARTICULATION_DEVICE_NAME        = REMOTE_SENSING_MAST
 ARTICULATION_DEVICE_ANGLE       = (1.00988<rad>, 0.0849357<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = (AZIMUTH, ELEVATION)
END_GROUP                        = RSM_ARTICULATION_STATE_PARMS

/* ARTICULATION DEVICE STATE: ROBOTIC ARM AT THE END (DP CREATION) */

GROUP                            = ARM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (1e+30<rad>, 1e+30<rad>, 1e+30<rad>, 1e+30<rad>, 1e+30<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                   "JOINT 2 SHOULDER ELEVATION",
                                   "JOINT 3 ELBOW-ENCODER",
                                   "JOINT 4 WRIST-ENCODER",
                                   "JOINT 5 TURRET-ENCODER")
 CONTACT_SENSOR_STATE            = OPEN
```

```
 CONATCT_SENSOR_STATE_NAME       = APXS_SWITCH
END_GROUP                        = ARM_ARTICULATION_STATE_PARMS

/* DATA OBJECT DEFINITION DESCRIPTIONS */

/* APXS CWA HEADER OBJECT */

OBJECT                           = CWA_HEADER_TABLE
   NAME                          = CWA_HEADER
   INTERCHANGE_FORMAT            = BINARY
   ROWS                          = 1
   COLUMNS                       = 9
   ROW_BYTES                     = 34
   DESCRIPTION                   = "
    This table contains a science associated instrument
    data as observed by the Mars Science Laboratory (MSL)
    Alpha Particle X-ray Spectrometer (APXS).

    The complete column definitions are contained in an external file
    found in the LABEL directory of the archive volume.
    "
   ^STRUCTURE                    = "APXS_EDR_CWA_HEADER.FMT"
END_OBJECT                       = CWA_HEADER_TABLE

/* APXS SCIENCE OBJECT */

OBJECT                           = SCIENCE_TABLE
   NAME                          = APXS_SPECTRA
   INTERCHANGE_FORMAT            = BINARY
   ROWS                          = 1
   COLUMNS                       = 15
   ROW_BYTES                     = 2098
   DESCRIPTION                   = "
    This table contains a science associated instrument
    data as observed by the Mars Science Laboratory (MSL)
    Alpha Particle X-ray Spectrometer (APXS).

    The complete column definitions are contained in an external file
    found in the LABEL directory of the archive volume.
    "
   ^STRUCTURE                    = "APXS_EDR_SCIENCE.FMT"
END_OBJECT                       = SCIENCE_TABLE

/* ERROR CONTROL VALUE OBJECT */

OBJECT                           = ERROR_CONTROL_TABLE
    NAME                         = ERROR_CONTROL
    INTERCHANGE_FORMAT           = BINARY
    ROWS                         = 1
    ROW_BYTES                    = 4
    COLUMNS                      = 1

      OBJECT                     = COLUMN
        NAME                     = ERROR_CONTROL_VALUE
        DATA_TYPE                = MSB_UNSIGNED_INTEGER
        START_BYTE               = 1
        BYTES                    = 4
        DESCRIPTION              = “Contains a CRC or Fletcher checksum value
                                   as indicated in the CHMN_HSKN_HEADER_TABLE
                                   control and status error control type field.”
      END_OBJECT                 = COLUMN
END_OBJECT                       = ERROR_CONTROL_TABLE
END
```

A.2 APXS_EDR_CWA_HEADER.FMT

```
/* APXS CWA EDR HEADER OBJECT */

OBJECT              = COLUMN
   NAME             = CONTACT_SWITCH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 1
   BYTES            = 4
   DESCRIPTION      = "APXS contact switch value.
                       0=OFF, 1=ON"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = NOT_USED
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 5
   BYTES            = 4
   DESCRIPTION      = "Temperature of the turret
                       at data product creation."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = GROUP_ID
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 9
   BYTES            = 2
   DESCRIPTION      = "Identification number of the
                       CYCLE_START data product containing
                       rover state information."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CMD_REPLY_FRAME_LENGTH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 11
   BYTES            = 4
   DESCRIPTION      = "APXS instrument Command Reply Frame length
                       length in bytes."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = CMD_REPLY_CONTROL_AND_STATUS
  DATA_TYPE         = MSB_UNSIGNED_INTEGER
  START_BYTE        = 15
  BYTES             = 4

    OBJECT          = BIT_COLUMN
      NAME          = OPCODE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 1
      BITS          = 8
      DESCRIPTION   = "Opcode."
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
```

```
    NAME           = ERROR_CONTROL_TYPE
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 9
    BITS           = 2
    DESCRIPTION    = "The type of error control algorithm applied by
                      the Instrument Transfer Frame. Valid:
                      0=reserved.
                      1=The CRC error control algorithm was applied
                      and placed in the error control field at the.
                      end of the frame.
                      2= The fletcher checksum error control algorithm
                      was applied and placed in the error control field
                      at the end of the frame.
                      3=No error control algorithm was applied and no
                      error control field is included."
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = DATA_PRESENT_FLAG
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 11
    BITS           = 1
    DESCRIPTION    = "Indicate the presence or absence of both the
                     Header Field Data Length and the Instrument
                     Transfer Frame Data Field. A value of '0'
                     indicates the Header Field Data Length and the
                     Instrument Transfer Frame Data Field are not
                     present; a value of '1' indicates the Header
                     Field Data Length and Instrument Transfer Frame
                     Data Field are present."
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = FRAME_TYPE
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 12
    BITS           = 1
    DESCRIPTION    = "Frame type.  0=Science Data Frame
                      1=Command Reply Frame"
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = AVERAGE_X_RAY_RATE
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 13
    BITS           = 3
    DESCRIPTION    = "Indicates the average X-Ray Rate. Valid in
                      Acquisition mode or thereafter:
                      Valid:
                      0: no X-Ray data detected
                      1: 1<=rate<10, sensor not properly deployed
                         (atmospheric rate 3)
                      2: 10<=rate<50, Sensor head far from sample
                      3: 50<=rate<120, large offset, similar to MER
                      4: 120<=rate<300,likely nominal measurements
                      5: 300<=rate<600, Sensor head very close to sample
                      6: 600<=rate<1500, very close or strange sample
```

```
                    7: rate>= 1500, very likely noisy detector"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = HEALTH_CHECK_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 16
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed its self
                     test.  Updated every 30 seconds.
                     Valid: 0=OK, 1=FAILED"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_EEPROM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 17
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     EEPROM check. The default upload parameter or the
                     Software-computed checksum differs from the embedded
                     checksum. 0=successful, 1=failed (default on
                     power up."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_RAM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 18
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     RAM check. 0=successful, 1=Failed"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COOLER_COMMAND_MODE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 19
    BITS          = 2
    DESCRIPTION   = "Indicates that mode of the cooler command. Valid:
                     0 = Off,
                     1 = Manual on
                     2 = auto mode"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = TIME_SYNC_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 21
    BITS          = 1
    DESCRIPTION   = "Indicates the Time Sync command status.
                    Valid: 0=received,
                    1=Not received since power up, send time sync command."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = WATCHDOG_RESET
```

```
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 22
    BITS          = 1
    DESCRIPTION   = "Identifies whether any un-commanded reboot happened in
                       the last instrument power cycle.  Information
                       updated only after each RCE communication with the
                       instrument.
                       Valid: 0=none, 1=yes."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = X_RAY-COUNT_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 23
    BITS          = 1
    DESCRIPTION   = "Status flag indicating whether any X-Ray signal were
                     detected in the last acquisition.  Information
                     updated only after each RCE communication with the
                     instrument. Valid: 0=none, 1=present"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = ACQUISITION_STATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 24
    BITS          = 1
    DESCRIPTION   = "Indicates the state of the instrument. Valid:
                    0 = Idle
                    1 = Acquiring"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COOLER_STATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 25
    BITS          = 1
    DESCRIPTION   = "Indicates the state of the APXS cooler. Updated
                    every 30 seconds. Valid: 0=off, 1=On"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = BOOT_STATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 26
    BITS          = 2
    DESCRIPTION   = "Indicates which software bank the instrument is
                     running from. Valid:
                     0=Gold Image 1
                     1=Gold Image 2
                     2=Silver Image 1
                     3=Silver Image 1"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COMMAND_CONDITION_CODE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 28
```

```
      BITS          = 5
      DESCRIPTION   = "Frame command condition code.  Valid: 0=reserved;
                      1=nominal execution; 2=Indicates a framing error
                      in the received command.  For a UART port, this
                      indicates that a byte was received without a
                      valid stop bit.  For HSS instruments, this
                      indicates that a frame signal was removed and a
                      non-integer number of bytes was received. 3=unused.
                      4=For UART port only, indicates that the recipient
                      detected bad parity on a received byte in a
                      command.5=unused; 6=Op code not defined for this
                      instrument; 7=Instrument does not support
                      specified error control type; 8=unused;
                      9="Reserved" Status Flags are unexpected.
                      10=condition code is not meaningful; 11=Data
                      present was set, but data length was missing or
                      zero, or the frame length was inconsistent with
                      the data length field. 12=checksum or CRC failed;
                      13=instrument in the wrong state or mode to
                      process this command. 14=Invalid, inconsistent or
                      unexpected data for this opcode; 15=unused;
                      16=Embedded and calculated checksums don't match in
                      the science data of UPLOAD_SOFTWARE cmd. 17=Embedded
                      and calculated checksums don't match in the science
                      data of UPLOAD_PARAMS command. 18-20=unused.
                      21=Unexpected value. 22=unused. 23=A command and/or
                      argument is invalid for the software image currently
                      running."
    END_OBJECT      = BIT_COLUMN
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CMD_REPLY_DATA_LENGTH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 19
   BYTES            = 4
   DESCRIPTION      = "APXS instrument Command Reply
                      frame data length in bytes."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SCIENCE_FRAME_LENGTH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 23
   BYTES            = 4
   DESCRIPTION      = "APXS instrument Science Frame
                      data length in bytes."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SCI_FRM_CONTROL_AND_STATUS
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 27
   BYTES            = 4

    OBJECT          = BIT_COLUMN
      NAME          = OPCODE
```

```
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 1
    BITS          = 8
    DESCRIPTION   = "Opcode."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = ERROR_CONTROL_TYPE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 9
    BITS          = 2
    DESCRIPTION   = "The type of error control algorithm applied by
                     the Instrument Transfer Frame. Valid:
                     0=reserved.
                     1=The CRC error control algorithm was applied
                     and placed in the error control field at the.
                     end of the frame.
                     2= The fletcher checksum error control algorithm
                     was applied and placed in the error control field
                     at the end of the frame.
                     3=No error control algorithm was applied and no
                     error control field is included."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = DATA_PRESENT_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 11
    BITS          = 1
    DESCRIPTION   = "Indicate the presence or absence of both the
                    Header Field Data Length and the Instrument
                    Transfer Frame Data Field. A value of '0'
                    indicates the Header Field Data Length and the
                    Instrument Transfer Frame Data Field are not
                    present; a value of '1' indicates the Header
                    Field Data Length and Instrument Transfer Frame
                    Data Field are present."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = FRAME_TYPE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 12
    BITS          = 1
    DESCRIPTION   = "Frame type.  0=Science Data Frame
                     1=Command Reply Frame"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = AVERAGE_X_RAY_RATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 13
    BITS          = 3
    DESCRIPTION   = "Indicates the average X-Ray Rate. Valid in
                     Acquisition mode or thereafter:
                     Valid:
                     0: no X-Ray data detected
```

```
                    1: 1<=rate<10, sensor not properly deployed
                       (atmospheric rate 3)
                    2: 10<=rate<50, Sensor head far from sample
                    3: 50<=rate<120, large offset, similar to MER
                    4: 120<=rate<300,likely nominal measurements
                    5: 300<=rate<600, Sensor head very close to sample
                    6: 600<=rate<1500, very close or strange sample
                    7: rate>= 1500, very likely noisy detector"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = HEALTH_CHECK_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 16
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed its self
                     test.  Updated every 30 seconds.
                     Valid: 0=OK, 1=FAILED"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_EEPROM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 17
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     EEPROM check. The default upload parameter or the
                     Software-computed checksum differs from the embedded
                     checksum. 0=successful, 1=failed (default on
                     power up."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_RAM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 18
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     RAM check. 0=successful, 1=Failed"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COOLER_COMMAND_MODE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 19
    BITS          = 2
    DESCRIPTION   = "Indicates that mode of the cooler command. Valid:
                     0 = Off,
                     1 = Manual on
                     2 = auto mode"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = TIME_SYNC_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 21
    BITS          = 1
```

```
  DESCRIPTION   = "Indicates the Time Sync command status.
                  Valid: 0=received,
                  1=Not received since power up, send time sync command."
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = WATCHDOG_RESET
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 22
  BITS          = 1
  DESCRIPTION   = "Identifies whether any un-commanded reboot happened in
                     the last instrument power cycle.  Information
                     updated only after each RCE communication with the
                     instrument.
                     Valid: 0=none, 1=yes."
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = X_RAY-COUNT_FLAG
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 23
  BITS          = 1
  DESCRIPTION   = "Status flag indicating whether any X-Ray signal were
                   detected in the last acquisition.  Information
                   updated only after each RCE communication with the
                   instrument. Valid: 0=none, 1=present"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = ACQUISITION_STATE
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 24
  BITS          = 1
  DESCRIPTION   = "Indicates the state of the instrument. Valid:
                  0 = Idle
                  1 = Acquiring"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = COOLER_STATE
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 25
  BITS          = 1
  DESCRIPTION   = "Indicates the state of the APXS cooler. Updated
                  every 30 seconds. Valid: 0=off, 1=On"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = BOOT_STATE
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 26
  BITS          = 2
  DESCRIPTION   = "Indicates which software bank the instrument is
                   running from. Valid:
                   0=Gold Image 1
                   1=Gold Image 2
                   2=Silver Image 1
```

```
                        3=Silver Image 1"
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = COMMAND_CONDITION_CODE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 28
      BITS          = 5
      DESCRIPTION   = "Frame command condition code.  Valid: 0=reserved;
                       1=nominal execution; 2=Indicates a framing error
                       in the received command.  For a UART port, this
                       indicates that a byte was received without a
                       valid stop bit.  For HSS instruments, this
                       indicates that a frame signal was removed and a
                       non-integer number of bytes was received. 3=unused.
                       4=For UART port only, indicates that the recipient
                       detected bad parity on a received byte in a
                       command.5=unused; 6=Op code not defined for this
                       instrument; 7=Instrument does not support
                       specified error control type; 8=unused;
                       9="Reserved" Status Flags are unexpected.
                       10=condition code is not meaningful; 11=Data
                       present was set, but data length was missing or
                       zero, or the frame length was inconsistent with
                       the data length field. 12=checksum or CRC failed;
                       13=instrument in the wrong state or mode to
                       process this command. 14=Invalid, inconsistent or
                       unexpected data for this opcode; 15=unused;
                       16=Embedded and calculated checksums don't match in
                       the science data of UPLOAD_SOFTWARE cmd. 17=Embedded
                       and calculated checksums don't match in the science
                       data of UPLOAD_PARAMS command. 18-20=unused.
                       21=Unexpected value. 22=unused. 23=A command and/or
                       argument is invalid for the software image currently
                       running."
    END_OBJECT      = BIT_COLUMN
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SCIENCE_FRAME_DATA_LEN
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 31
   BYTES            = 4
   DESCRIPTION      = "Science frame data length in bytes."
END_OBJECT          = COLUMN
```

A.3 APXS_EDR_SCI_HEADER.FMT

```
/* APXS SCIENCE HEADER TABLE OBJECT */

OBJECT              = COLUMN
   NAME             = CONTACT_SWITCH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 1
   BYTES            = 4
   DESCRIPTION      = "APXS contact switch value.
                      0=OFF, 1=ON"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = NOT_USED
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 5
   BYTES            = 4
   DESCRIPTION      = "SCLK at APXS power on or
                      when rover last woke up in
                      seconds."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = NOT_USED2
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 9
   BYTES            = 4
   DESCRIPTION      = "Temperature of the turret
                      before acquisition."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = DP_TURRET_TEMP
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 13
   BYTES            = 4
   DESCRIPTION      = "Temperature of the turret
                      at data product creation. 0=
                      Not available."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = GROUP_ID
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 17
   BYTES            = 2
   DESCRIPTION      = "Identification number of the
                      CYCLE_START data product containing
                      rover state information."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CMD_REPLY_FRAME_LENGTH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 19
```

```
   BYTES            = 4
   DESCRIPTION      = "APXS instrument Command Reply
                      Frame length in bytes."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
  NAME              = CMD_REPLY_CONTROL_AND_STATUS
  DATA_TYPE         = MSB_UNSIGNED_INTEGER
  START_BYTE        = 23
  BYTES             = 4

    OBJECT          = BIT_COLUMN
      NAME          = OPCODE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 1
      BITS          = 8
      DESCRIPTION   = "Opcode."
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = ERROR_CONTROL_TYPE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 9
      BITS          = 2
      DESCRIPTION   = "The type of error control algorithm applied by
                       the Instrument Transfer Frame. Valid is
                       0=reserved.
                       1=The CRC error control algorithm was applied
                       and placed in the error control field at the.
                       end of the frame.
                       2= The fletcher checksum error control algorithm
                       was applied and placed in the error control field
                       at the end of the frame.
                       3=No error control algorithm was applied and no
                       error control field is included."
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = DATA_PRESENT_FLAG
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 11
      BITS          = 1
      DESCRIPTION   = "Indicate the presence or absence of both the
                      Header Field Data Length and the Instrument
                      Transfer Frame Data Field. A value of 0
                      indicates the Header Field Data Length and the
                      Instrument Transfer Frame Data Field are not
                      present; a value of 1 indicates the Header
                      Field Data Length and Instrument Transfer Frame
                      Data Field are present."
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = FRAME_TYPE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 12
      BITS          = 1
```

```
    DESCRIPTION   = "Frame type.  0=Science Data Frame
                     1=Command Reply Frame"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = AVERAGE_X_RAY_RATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 13
    BITS          = 3
    DESCRIPTION   = "Indicates the average X-Ray Rate. Valid in
                     Acquistion mode or thereafter:
                     Valid:
                     0: no X-Ray data detected
                     1: 1<=rate<10, sensor not properly deployed
                        (atmospheric rate 3)
                     2: 10<=rate<50, Sensor head far from sample
                     3: 50<=rate<120, large offset, similar to MER
                     4: 120<=rate<300,likely nominal measurements
                     5: 300<=rate<600, Sensor head very close to sample
                     6: 600<=rate<1500, very close or strange sample
                     7: rate>= 1500, very likely noisy detector"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = HEALTH_CHECK_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 16
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed its self
                     test.  Updated every 30 seconds.
                     Valid: 0=OK, 1=FAILED"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_EEPROM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 17
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     EEPROM check. The default upload parameter or the
                     Software-computed checksum differs from the embedded
                     checksum. 0=successful, 1=failed (default on
                     power up."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = POST_RAM_FAIL_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 18
    BITS          = 1
    DESCRIPTION   = "Indicates if the instrument has failed the post
                     RAM check. 0=successful, 1=Failed"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COOLER_COMMAND_MODE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
```

```
    START_BIT     = 19
    BITS          = 2
    DESCRIPTION   = "Indicates that mode of the cooler command. Valid:
                     0 = Off,
                     1 = Manual on
                     2 = auto mode"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = TIME_SYNC_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 21
    BITS          = 1
    DESCRIPTION   = "Indicates the Time Sync command status.
                    Valid: 0=received,
                    1=Not received since power up, send time sync command."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = WATCHDOG_RESET
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 22
    BITS          = 1
    DESCRIPTION   = "Identifies whether any un-commanded reboot happened in
                       the last instrument power cycle.  Information
                       updated only after each RCE communication with the
                       instrument.
                       Valid: 0=none, 1=yes."
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = X_RAY_COUNT_FLAG
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 23
    BITS          = 1
    DESCRIPTION   = "Status flag indicating whether any X-Ray signal were
                     detected in the last acquisition.  Information
                     updated only after each RCE communication with the
                     instrument. Valid: 0=none, 1=present"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = ACQUISITION_STATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 24
    BITS          = 1
    DESCRIPTION   = "Indicates the state of the instrument. Valid:
                    0 = Idle
                    1 = Acquiring"
  END_OBJECT      = BIT_COLUMN

  OBJECT          = BIT_COLUMN
    NAME          = COOLER_STATE
    BIT_DATA_TYPE = UNSIGNED_INTEGER
    START_BIT     = 25
    BITS          = 1
    DESCRIPTION   = "Indicates the state of the APXS cooler. Updated
```

```
                      every 30 seconds. Valid: 0=off, 1=On"
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = BOOT_STATE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 26
      BITS          = 2
      DESCRIPTION   = "Indicates which software bank the instrument is
                       running from. Valid:
                       0=Gold Image 1
                       1=Gold Image 2
                       2=Silver Image 1
                       3=Silver Image 1"
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = COMMAND_CONDITION_CODE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 28
      BITS          = 5
      DESCRIPTION   = "Frame command condition code.  Valid: 0=reserved;
                      1=nominal execution; 2=Indicates a framing error
                      in the received command.  For a UART port, this
                      indicates that a byte was received without a
                      valid stop bit.  For HSS instruments, this
                      indicates that a frame signal was removed and a
                      non-integer number of bytes was received. 3=unused.
                      4=For UART port only, indicates that the recipient
                      detected bad parity on a received byte in a
                      command.5=unused; 6=Op code not defined for this
                      instrument; 7=Instrument does not support
                      specified error control type; 8=unused;
                      9="Reserved" Status Flags are unexpected.
                      10=condition code is not meaningful; 11=Data
                      present was set, but data length was missing or
                      zero, or the frame length was inconsistent with
                      the data length field. 12=checksum or CRC failed;
                      13=instrument in the wrong state or mode to
                      process this command. 14=Invalid, inconsistent or
                      unexpected data for this opcode; 15=unused;
                      16=Embedded and calculated checksums don't match in
                      the science data of UPLOAD_SOFTWARE cmd. 17=Embedded
                      and calculated checksums don't match in the science
                      data of UPLOAD_PARAMS command. 18-20=unused.
                      21=Unexpected value. 22=unused. 23=A command and/or
                      argument is invalid for the software image currently
                      running."
    END_OBJECT      = BIT_COLUMN
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CMD_REPLY_DATA_LENGTH
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 27
   BYTES            = 4
   DESCRIPTION      = "APXS instrument Command Reply
```

```
                        frame data length in bytes."
END_OBJECT              = COLUMN

OBJECT                  = COLUMN
   NAME                 = SCIENCE_FRAME_LENGTH
   DATA_TYPE            = MSB_UNSIGNED_INTEGER
   START_BYTE           = 31
   BYTES                = 4
   DESCRIPTION          = "APXS instrument Science Frame
                          data length in bytes."
END_OBJECT              = COLUMN

OBJECT                  = COLUMN
   NAME                 = SCI_FRM_CONTROL_AND_STATUS
   DATA_TYPE            = MSB_UNSIGNED_INTEGER
   START_BYTE           = 35
   BYTES                = 4

    OBJECT              = BIT_COLUMN
      NAME              = OPCODE
      BIT_DATA_TYPE     = UNSIGNED_INTEGER
      START_BIT         = 1
      BITS              = 8
      DESCRIPTION       = "Opcode."
    END_OBJECT          = BIT_COLUMN

    OBJECT              = BIT_COLUMN
      NAME              = ERROR_CONTROL_TYPE
      BIT_DATA_TYPE     = UNSIGNED_INTEGER
      START_BIT         = 9
      BITS              = 2
      DESCRIPTION       = "The type of error control algorithm applied by
                           the Instrument Transfer Frame. Valid:
                           0=reserved.
                           1=The CRC error control algorithm was applied
                           and placed in the error control field at the.
                           end of the frame.
                           2= The fletcher checksum error control algorithm
                           was applied and placed in the error control field
                           at the end of the frame.
                           3=No error control algorithm was applied and no
                           error control field is included."
    END_OBJECT          = BIT_COLUMN

    OBJECT              = BIT_COLUMN
      NAME              = DATA_PRESENT_FLAG
      BIT_DATA_TYPE     = UNSIGNED_INTEGER
      START_BIT         = 11
      BITS              = 1
      DESCRIPTION       = "Indicate the presence or absence of both the
                          Header Field Data Length and the Instrument
                          Transfer Frame Data Field. A value of '0'
                          indicates the Header Field Data Length and the
                          Instrument Transfer Frame Data Field are not
                          present; a value of '1' indicates the Header
                          Field Data Length and Instrument Transfer Frame
                          Data Field are present."
```

```
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = FRAME_TYPE
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 12
  BITS          = 1
  DESCRIPTION   = "Frame type.  0=Science Data Frame
                   1=Command Reply Frame"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = AVERAGE_X_RAY_RATE
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 13
  BITS          = 3
  DESCRIPTION   = "Indicates the average X-Ray Rate. Valid in
                   Acquisition mode or thereafter:
                   Valid:
                   0: no X-Ray data detected
                   1: 1<=rate<10, sensor not properly deployed
                      (atmospheric rate 3)
                   2: 10<=rate<50, Sensor head far from sample
                   3: 50<=rate<120, large offset, similar to MER
                   4: 120<=rate<300,likely nominal measurements
                   5: 300<=rate<600, Sensor head very close to sample
                   6: 600<=rate<1500, very close or strange sample
                   7: rate>= 1500, very likely noisy detector"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = HEALTH_CHECK_FLAG
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 16
  BITS          = 1
  DESCRIPTION   = "Indicates if the instrument has failed its self
                   test.  Updated every 30 seconds.
                   Valid: 0=OK, 1=FAILED"
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = POST_EEPROM_FAIL_FLAG
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 17
  BITS          = 1
  DESCRIPTION   = "Indicates if the instrument has failed the post
                   EEPROM check. The default upload parameter or the
                   Software-computed checksum differs from the embedded
                   checksum. 0=successful, 1=failed (default on
                   power up."
END_OBJECT      = BIT_COLUMN

OBJECT          = BIT_COLUMN
  NAME          = POST_RAM_FAIL_FLAG
  BIT_DATA_TYPE = UNSIGNED_INTEGER
  START_BIT     = 18
  BITS          = 1
```

```
    DESCRIPTION    = "Indicates if the instrument has failed the post
                      RAM check. 0=successful, 1=Failed"
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = COOLER_COMMAND_MODE
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 19
    BITS           = 2
    DESCRIPTION    = "Indicates that mode of the cooler command. Valid:
                      0 = Off,
                      1 = Manual on
                      2 = auto mode"
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = TIME_SYNC_FLAG
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 21
    BITS           = 1
    DESCRIPTION    = "Indicates the Time Sync command status.
                     Valid: 0=received,
                     1=Not received since power up, send time sync command."
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = WATCHDOG_RESET
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 22
    BITS           = 1
    DESCRIPTION    = "Identifies whether any un-commanded reboot happened in
                        the last instrument power cycle.  Information
                        updated only after each RCE communication with the
                        instrument.
                        Valid: 0=none, 1=yes."
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = X_RAY-COUNT_FLAG
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 23
    BITS           = 1
    DESCRIPTION    = "Status flag indicating whether any X-Ray signal were
                      detected in the last acquisition.  Information
                      updated only after each RCE communication with the
                      instrument. Valid: 0=none, 1=present"
  END_OBJECT       = BIT_COLUMN

  OBJECT           = BIT_COLUMN
    NAME           = ACQUISITION_STATE
    BIT_DATA_TYPE  = UNSIGNED_INTEGER
    START_BIT      = 24
    BITS           = 1
    DESCRIPTION    = "Indicates the state of the instrument. Valid:
                     0 = Idle
                     1 = Acquiring"
  END_OBJECT       = BIT_COLUMN
```

```
    OBJECT          = BIT_COLUMN
      NAME          = COOLER_STATE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 25
      BITS          = 1
      DESCRIPTION   = "Indicates the state of the APXS cooler. Updated
                      every 30 seconds. Valid: 0=off, 1=On"
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = BOOT_STATE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 26
      BITS          = 2
      DESCRIPTION   = "Indicates which software bank the instrument is
                       running from. Valid:
                       0=Gold Image 1
                       1=Gold Image 2
                       2=Silver Image 1
                       3=Silver Image 1"
    END_OBJECT      = BIT_COLUMN

    OBJECT          = BIT_COLUMN
      NAME          = COMMAND_CONDITION_CODE
      BIT_DATA_TYPE = UNSIGNED_INTEGER
      START_BIT     = 28
      BITS          = 5
      DESCRIPTION   = "Frame command condition code.  Valid: 0=reserved;
                       1=nominal execution; 2=Indicates a framing error
                       in the received command.  For a UART port, this
                       indicates that a byte was received without a
                       valid stop bit.  For HSS instruments, this
                       indicates that a frame signal was removed and a
                       non-integer number of bytes was received. 3=unused.
                       4=For UART port only, indicates that the recipient
                       detected bad parity on a received byte in a
                       command.5=unused; 6=Op code not defined for this
                       instrument; 7=Instrument does not support
                       specified error control type; 8=unused;
                       9="Reserved" Status Flags are unexpected.
                       10=condition code is not meaningful; 11=Data
                       present was set, but data length was missing or
                       zero, or the frame length was inconsistent with
                       the data length field. 12=checksum or CRC failed;
                       13=instrument in the wrong state or mode to
                       process this command. 14=Invalid, inconsistent or
                       unexpected data for this opcode; 15=unused;
                       16=Embedded and calculated checksums don't match in
                       the science data of UPLOAD_SOFTWARE cmd. 17=Embedded
                       and calculated checksums don't match in the science
                       data of UPLOAD_PARAMS command. 18-20=unused.
                       21=Unexpected value. 22=unused. 23=A command and/or
                       argument is invalid for the software image currently
                       running."
    END_OBJECT      = BIT_COLUMN
END_OBJECT          = COLUMN
```

```
OBJECT              = COLUMN
   NAME             = SCIENCE_FRAME_DATA_LEN
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 39
   BYTES            = 4
   DESCRIPTION      = "Science frame data length in bytes."
END_OBJECT          = COLUMN
```

A.4 APXS_EDR_SCIENCE.FMT

```
/* APXS MEASUREMENT TABLE OBJECT */

OBJECT              = COLUMN
   NAME             = SUM_COUNTER
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 1
   BYTES            = 4
   DESCRIPTION      = "Spectrum sum counter."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = START_TIME
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 5
   BYTES            = 4
   DESCRIPTION      = "Absolute start time in SCLK seconds with respect
                      Respect to the last time sync or clock second
                      Counter in case no time sync was
                      received."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = STOP_TIME
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 9
   BYTES            = 4
   DESCRIPTION      = "Absolute stop time in SCLK seconds with respect
                      to the absolute start time."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = MAIN_ELECTRONICS_TEMP_SUM
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 13
   BYTES            = 4
   DESCRIPTION      = "Summation of the main electronics temperatures
                      in raw counts.
                      Degrees Celcius=(HEX2DEC(value)/228.72)-165"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = MAIN_SENSOR_HEAD_TEMP_SUM
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 17
   BYTES            = 4
   DESCRIPTION      = "Summation of the main sensor head temperatures
                      in raw counts.
                      Degrees Celcius=(HEX2DEC(value)/190.3)-250"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = MAIN_ELECTRONICS_TEMP_COUNT
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 21
```

```
   BYTES              = 2
   DESCRIPTION        = "Total number of the main electronics
                        Temperature measurements.
                        Degrees Celcius=(HEX2DEC(value)/228.72)-165"
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = MAIN_SENSOR_HEAD_TEMP_COUNT
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 23
   BYTES              = 2
   DESCRIPTION        = "Total number of the main sensor head
                        temperature measurements.
                        Degrees Celcius=(HEX2DEC(value)/190.3)-250"
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = MAIN_ELECTRONICS_TEMP_MIN
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 25
   BYTES              = 4
   DESCRIPTION        = "Minimum temperature of the main
                        electronics in raw counts.
                        Degrees Celcius=(HEX2DEC(value)/228.72)-165"
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = MAIN_ELECTRONICS_TEMP_MAX
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 29
   BYTES              = 4
   DESCRIPTION        = "Maximum temperature of the main
                        electronics in raw counts.
                        Degrees Celcius=(HEX2DEC(value)/228.72)-165"
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = MAIN_SENSOR_HEAD_TEMP_MIN
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 33
   BYTES              = 4
   DESCRIPTION        = "Minimum temperature of the sensor
                        head in raw counts.
                        Degrees Celcius=(HEX2DEC(value)/190.3)-250"

END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = MAIN_SENSOR_HEAD_TEMP_MAX
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 37
   BYTES              = 4
   DESCRIPTION        = "Maximum temperature of the sensor
                        Head in raw counts.
                        Degrees Celcius=(HEX2DEC(value)/190.3)-250"

END_OBJECT            = COLUMN
```

```
OBJECT              = COLUMN
   NAME             = HIGH_VOLTAGE_RAIL
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 41
   BYTES            = 4
   DESCRIPTION      = "High voltage rail measured at the end
                      of acquisition in raw counts.
                      Voltage = -(HEX2DEC(value)/65535)*182.995"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = BACK_VOLTAGE_BIAS
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 45
   BYTES            = 4
   DESCRIPTION      = "Back voltage bias measured at the end
                      of acquisition in raw counts.
                      Voltage = -(HEX2DEC(value)/65535)*182.995"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = XRAY_COUNTS
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 49
   BYTES            = 2048
   ITEMS            = 1024
   ITEM_BYTES       = 2
   ITEM_OFFSET      = 2
   DESCRIPTION      = "Number of x-rays detected per
                       channel. Channels represent
                       different energy levels."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = DEAD_TIME
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2097
   BYTES            = 2
   DESCRIPTION      = "Dead Time in in 1/10 seconds"
END_OBJECT          = COLUMN
```

A.5 APXS_EDR_ENGINEERING.FMT

```
/* APXS ENGINEERING TABLE OBJECT */

OBJECT              = COLUMN
   NAME             = LAST_TIME_SYNC
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 1
   BYTES            = 4
   DESCRIPTION      = "Time sync last received in SCLK
                      Seconds.  Zero if no time sync received."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = RESERVED
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 5
   BYTES            = 16
   ITEMS            = 16
   ITEM_BYTES       = 1
   ITEM_OFFSET      = 1
   DESCRIPTION      = "Reserved bytes."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = VALID_COMMAND_COUNT
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 21
   BYTES            = 4
   DESCRIPTION      = "Number of valid commands since
                      instrument power on."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = REJECTED_COMMAND_COUNT
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 25
   BYTES            = 4
   DESCRIPTION      = "Number of rejected commands since
                      instrument power on."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SOFTWARE_VERSION
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 29
   BYTES            = 4
   DESCRIPTION      = "Software version used.
                      0=Gold Image 1, 1=Gold Image 2, 3=Silver Image 1,
                      4=Silver Image 2."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SOFTWARE_CHECKSUM
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 33
```

```
   BYTES              = 4
   DESCRIPTION        = "Flight software checksum."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = CYCLE_TIME
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 37
   BYTES              = 4
   DESCRIPTION        = "Requested fixed time span for data
                        accumulation cycle in seconds.
                        Valid range is 1 - 10800."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = TEMPERATURE_LOOKUP_TABLE
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 41
   BYTES              = 2380
   ITEMS              = 1190
   ITEM_BYTES         = 2
   ITEM_OFFSET        = 2
   DESCRIPTION        = "Temperature Lookup Table used
                        for acquired data calibration."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COOLER_ON_THRESHOLD
   DATA_TYPE          = MSB_INTEGER
   START_BYTE         = 2421
   BYTES              = 4
   DESCRIPTION        = "Temperature threshold value in degrees
                        Celsius for automatic power on of the
                        cooler. Valid range is -130 to +20."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COOLER_OFF_THRESHOLD
   DATA_TYPE          = MSB_INTEGER
   START_BYTE         = 2425
   BYTES              = 4
   DESCRIPTION        = "Temperature threshold value in degrees
                        Celsius for automatic power off of the
                        cooler. Valid range is -130 to +20."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COMPARATOR_THRESHOLD_VOLTAGE
   DATA_TYPE          = MSB_UNSIGNED_INTEGER
   START_BYTE         = 2429
   BYTES              = 2
   DESCRIPTION        = "Value for controlling the
                        comparator threshold voltage.
                        Valid range is 0 to 4095."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
```

```
   NAME              = HIGH_VOLTAGE_LOGIC_FLAG
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2431
   BYTES             = 1
   DESCRIPTION       = "0=Negative logic enabled,
                       1=Positive Logic enabled."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = MAIN_ELECTRONIC_TEMP_HIGH_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2432
   BYTES             = 1
   DESCRIPTION       = "High end limit setting for the
                      Main electronics temperature."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = MAIN_ELECTRONIC_TEMP_LOW_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2433
   BYTES             = 1
   DESCRIPTION       = "Low end limit setting for the
                       main electronics temperature."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = SENSOR_HEAD_TEMP_HIGH_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2434
   BYTES             = 1
   DESCRIPTION       = "High end limit setting for the
                      sensor head temperature."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = HIGH_VOLTAGE_RAIL_HIGH_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2435
   BYTES             = 1
   DESCRIPTION       = "High end limit setting for the
                      High voltage rail."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = HIGH_VOLTAGE_RAIL_LOW_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2436
   BYTES             = 1
   DESCRIPTION       = "Low end limit setting for the
                      high voltage rail."
END_OBJECT           = COLUMN

OBJECT               = COLUMN
   NAME              = CABLE_RESISTANCE_HIGH_LIMIT
   DATA_TYPE         = UNSIGNED_INTEGER
   START_BYTE        = 2437
```

```
   BYTES              = 1
   DESCRIPTION        = "High end limit setting for the
                        cable resistance."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = CABLE_RESISTANCE_LOW_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2438
   BYTES              = 1
   DESCRIPTION        = "Low end limit setting for the
                        cable resistance."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COMPARATOR_THRESHOLD_HIGH_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2439
   BYTES              = 1
   DESCRIPTION        = "High end limit setting for the
                        comparator threshold voltage."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COMPARATOR_THRESHOLD_LOW_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2440
   BYTES              = 1
   DESCRIPTION        = "Low end limit setting for the
                         Comparator threshold voltage."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = BACK_VOLTAGE_BIAS_HIGH_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2441
   BYTES              = 1
   DESCRIPTION        = "High end limit setting for the
                        back voltage bias."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = BACK_VOLTAGE_BIAS_LOW_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2442
   BYTES              = 1
   DESCRIPTION        = "Low end limit setting for the
                        back voltage bias."
END_OBJECT            = COLUMN

OBJECT                = COLUMN
   NAME               = COOLER_VOLTAGE_HIGH_LIMIT
   DATA_TYPE          = UNSIGNED_INTEGER
   START_BYTE         = 2443
   BYTES              = 1
   DESCRIPTION        = "High end limit setting for the
                        cooler power voltage."
```

```
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = COOLER_VOLTAGE_LOW_LIMIT
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2444
   BYTES            = 1
   DESCRIPTION      = "Low end limit setting for the
                      cooler power voltage."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = RESERVED
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2445
   BYTES            = 8
   DESCRIPTION      = "Reserved bytes"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = PARAMETER_CHECKSUM
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2453
   BYTES            = 4
   DESCRIPTION      = "Checksum of parameters requested."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = MAIN_ELECTRONICS_TEMP
   DATA_TYPE        = MSB_INTEGER
   START_BYTE       = 2457
   BYTES            = 4
   DESCRIPTION      = "Main electronics
                      temperature.
                      Degrees Celcius=(HEX2DEC(value)/228.72)-165"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SENSOR_HEAD_TEMP
   DATA_TYPE        = MSB_INTEGER
   START_BYTE       = 2461
   BYTES            = 4
   DESCRIPTION      = "Sensor head temperature.
                      Degrees Celcius=(HEX2DEC(value)/190.3)-250"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = HIGH_VOLTAGE_RAIL
   DATA_TYPE        = MSB_INTEGER
   START_BYTE       = 2465
   BYTES            = 4
   DESCRIPTION      = "High voltage rail.
                      Voltage=(HEX2DEC(value)/65535)*182.995"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CABLE_RESISTANCE
```

```
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2469
   BYTES            = 4
   DESCRIPTION      = "Cable resistance.
                      mOhm units=(HEX2DEC(value)/.3408)"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = COMPARATOR_THRESHOLD_VOLTAGE
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2473
   BYTES            = 4
   DESCRIPTION      = "Comparator threshold voltage.
                      Voltage=(HEX2DEC(value)/65535)*32.764"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = BACK_VOLTAGE_BIAS
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2477
   BYTES            = 4
   DESCRIPTION      = "Back voltage bias.
                      Voltage=(HEX2DEC(value)/65535)*182.995"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = COOLER_POWER_VOLTAGE
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2481
   BYTES            = 4
   DESCRIPTION      = "Cooler power voltage.
                      Voltage=HEX2DEC(value)/16383.75"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SOFTWARE_CHECKSUM_2
   DATA_TYPE        = MSB_UNSIGNED_INTEGER
   START_BYTE       = 2485
   BYTES             = 4
   DESCRIPTION      = "Flight software redundant checksum."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = MAIN_ELECTRONICS_TEMP_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2489
   BYTES            = 1
   DESCRIPTION      = "Main electronics unit temperature
                      out of range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SENSOR_HEAD_TEMP_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2490
   BYTES            = 1
```

```
   DESCRIPTION      = "Sensor head temperature
                      out of range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = HIGH_VOLTAGE_RAIL_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2491
   BYTES            = 1
   DESCRIPTION      = "High voltage rail out of
                      range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = CABLE_RESISTANCE_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2492
   BYTES            = 1
   DESCRIPTION      = "Cable resistance out of
                      range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = COMPARATOR_THRESHOLD_VOLTAGE_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2493
   BYTES            = 1
   DESCRIPTION      = "Comparator threshold voltage
                      out of range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = BACK_VOLTAGE_BIAS_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2494
   BYTES            = 1
   DESCRIPTION      = "Back voltage bias out of
                      range test. 0=pass,
                       1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = COOLER_POWER_VOLTAGE_RANGE_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2495
   BYTES            = 1
   DESCRIPTION      = "Cooler power voltage out of
                      range test. 0=pass,
                      1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = SPARE_FLAG
```

```
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2496
   BYTES            = 1
   DESCRIPTION      = "Spare out of range test.
                      0=pass, 1=out of range."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = POST_RAM_CHECK_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2497
   BYTES            = 1
   DESCRIPTION      = "POST RAM check test.
                      0=pass, 1=fail."
END_OBJECT          = COLUMN

OBJECT              = COLUMN
   NAME             = POST_EEPROM_CHECK_FLAG
   DATA_TYPE        = UNSIGNED_INTEGER
   START_BYTE       = 2498
   BYTES            = 1
   DESCRIPTION      = "POST EEPROM check test.
                       0=pass, 1=fail."
END_OBJECT          = COLUMN
```

APPENDIX B – APXS LABEL KEYWORD DEFINITIONS

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Ops Keyword ACTIVE_FLIGHT_STRING_ID **PDS Keyword** MSL:ACTIVE_FLIGHT_STRING_ID **Definition** Indicates which flight computer string was active when this product was acquired.	**Valid Values** “A”, “B” **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** “MslEarthProductMetadata:MslProductMetadata:CreationStringId” **Type** U16
Ops Keyword APPLICATION_PROCESS_ID **Definition** Specifies the name associated with the source or process which created the data. For MSL, the Application Process Identifier (APID) identifies the data type encapsulated in the packet. The APIDS can vary between each version of FSW. As of FSW version **TBD**, integer APIDs will be static and remain unchanged. However, only APID Names uniquely identify Data Product types across all FSW versions. See also APPLICATION_PROCESS_NAME and Appendix C.	**Valid Values** (as of FSW version **10.5.7**) APXS APID Name “**94**” ApxsCwa “**97**” ApxsScienceAndEng **Type** string(256) **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** “MslEarthProductMetadata:MslProductMetadata:Apid” **Type** n/a

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Ops Keyword APPLICATION_PROCESS_NAME **Definition** Specifies the name associated with the source or process which created the data. For MSL, the Application Process Identifier (APID) identifies the data type encapsulated in the packet. The APIDS can vary between each version of FSW. As of FSW version **TBD**, integer APIDs will be static and remain unchanged. However, only APID Names uniquely identify Data Product types across all FSW versions. See also APPLICATION_PROCESS_ID and Appendix C.	**Valid Values** Apxs Name APID "ApxsCwa" **94** "ApxsScienceAndEng" **97** **Type** string(256) **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:ProductName" **Type** n/a
Ops Keyword ARTICULATION_DEVICE_ANGLE **Definition** Specifies the value of an angle between two parts or segments of an articulated device. NOTE: MER used radians. The PDS default unit for this keyword is degrees, so the <rad> tag is required for MSL data.	**Valid Values** n/a **Type** float array[5] **Units** radians (<rad> unit tag required) **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) END_ARM_ARTICULATION_STATE (Group)	**Mode** 1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:SampleArmOrientation" **Type** 4) F32[5]

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Ops Keyword ARTICULATION_DEVICE_ANGLE_NAME **Definition** Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE.	**Valid Values** ARM ("JOINT 1 AZIMUTH-ENCODER ", "JOINT 2 ELEVATION-ENCODER ", "JOINT 3 ELBOW-ENCODER", "JOINT 4 WRIST-ENCODER", "JOINT 5 TURRET-ENCODER") for both START and END **Type** string array[5] **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** Static values **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword ARTICULATION_DEVICE_ID **Definition** 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 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.	**Valid Values** "ARM" for both START and END **Type** string **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** Static value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword ARTICULATION_DEVICE_NAME **Definition** 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.	**Valid Values** "SAMPLE ARM" for both START and END **Type** string **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** Static value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Ops Keyword ARTICULATION_DEVICE_TEMP **Definition** Specifies the temperature, in degrees Celsius, of an articulated device or some part of an articulated device.	**Valid Values** "-3.4e38" to "3.4e38" **Type** float array[1] **Units** deg C (<degC> unit tag required) **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** 1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "<ANCILLARY DPO>:turret_temp_dp_create" **Type** F32[5]
Ops Keyword ARTICULATION_DEVICE_TEMP_NAME **Definition** Specifies the array of formal names identifying each of the values used in ARTICULATION_DEVICE_TEMP.	**Valid Values** "TURRET JOINT" **Type** string array[2] **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** Static value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword AUTO_DELETE_FLAG **PDS Keyword** MSL:AUTO_DELETE_FLAG **Definition** Indicates if the DP will be deleted upon transmission: 1 for delete, 0 for no delete.	**Valid Values** "**0"="FALSE"** **"1"="TRUE"** **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:DeleteOnSend" **Type** n/a
Ops Keyword COMMAND_SEQUENCE_NUMBER	**Valid Values** n/a	**Mode** EMD in XML format

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies a numeric identifier for a sequence of commands sent to a spacecraft or instrument. NOTE: For MSL, this is the command number which identifies the specific generating command within the specified sequence.	**Type** integer **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:CommandNumber" **Type** n/a
Ops Keyword COMMUNICATION_SESSION_ID **PDS Keyword** MSL:COMMUNICATION_SESSION_ID **Definition** Active Communication Session ID at time of MPDU (Metadata Protocol Data Unit) creation. A Data Product is formed into PDUs (Protocol Data Units) by creating one MPDU, one or more PPDUs (Product Data PDUs), and one EPDU (End-of-Product PDU). The MPDU is the first PDU produced for a data product, and contains general and MSL specific "data about the data product". It is wholly contained in a single packet with the Packet Sequence Count always set to zero. (Above text from MSL FGICD, v2.2.1) See also EXPECTED_TRANSMISSION_PATH.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:CommSessionId" **Type** n/a
Ops Keyword CONTACT_SENSOR_STATE	**Valid Values** 0 = "CLOSED"	**Mode** 1) EMD in XML format from the ApxsStart data product where

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies an array of identifiers for the state of an instrument or an instrument host's contact sensors at a specified time.	1 = "OPEN" **Type** string array[1] **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "<ANCILLARY DPO>:contact_switch_dp_create" **Type** U32
Ops Keyword CONTACT_SENSOR_STATE_NAME **Definition** Specifies the possible value that can be contained in the CONTACT_SENSOR_STATE array.	**Valid Values** "APXS SWITCH" **Type** string array[1] **Units** n/a **Location** 1) START_ARM_ARTICULATION_STATE (Group) 2) ARM_ARTICULATION_STATE (Group)	**Mode** Static value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword COORDINATE_SYSTEM_INDEX **Definition** 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. Example: COORDINATE_SYSTEM_INDEX=(1,1) For MSL, see ROVER_MOTION_COUNTER and ROVER_MOTION_COUNTER_NAME for the 10 indices and names.	**Valid Values** n/a **Type** integer array[6] **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** 1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:RoverMotionCounter NOTES: • Defaults for EDRs: - If ROVER_FRAME or SITE_FRAME or MAST_FRAME or LOCAL_LEVEL_FRAME or DRILL_FRAME, then Site, Drive, Pose, Arm, CHIMRA, Drill, RSM, HGA, DRT and IC indices **Type** U16
Ops Keyword COORDINATE_SYSTEM_INDEX_NAME	**Valid Values** 1) "SITE"	**Mode** Static Values

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies an array of the formal names identifying each integer specified in COORDINATE_SYSTEM_INDEX.	2) “DRIVE” 3) “POSE” 4) “ARM” 5) “CHIMRA” 6) “DRILL” 7) “RSM” 8) “HGA” 9) “DRT” 10) “IC” **Type** string array[10] **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Field as “<xml name>:[<element>]:[<element>]:<field>”** NOTES: • Should match the number of values in COORDINATE_SYSTEM_INDEX. **Type** U16
Ops Keyword COORDINATE_SYSTEM_NAME **Definition** 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.	**Valid Values** “ROVER_FRAME” for START and END **Type** string **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** Static Value **Field as “<xml name>:[<element>]:[<element>]:<field>”** **Type**
Ops Keyword DATA_SET_ID	**Valid Values**	**Mode** PDS, Table Lookup

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition 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 of the DATA_SET_NAME. In the PDS, the values for DATA_SET_ID are constructed according to standards outlined in the Standards Reference.	"MSL-M-APXS-2-EDR-V1.0", **Type** string(40) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword DATA_SET_NAME **Definition** 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.	**Valid Values** "Operations" EDRs "MSL MARS ALPHA PARTICLE X-RAY SPECTROMETER EDR V1.0", **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** PDS, Table Lookup **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword DOWNLOAD_PRIORITY **Definition** Specifies which data to downlink/transmit, based on order of importance. The lower numerical priority (higher-ranked number) data products are transmitted before higher numerical priority (lower-ranked number) data products. For example, an image with a downlink priority of 25 will be transmitted from the rover before an image with a downlink priority of 50.	**Valid Values** "0" to "100" **Type** integer **Units** n/a **Location** TELEMETRY	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:ProductPriority" **Type** U8
Ops Keyword EARTH_RECEIVED_START_TIME	**Valid Values** YYYY-DOYThh:mm:ss[.fff]	**Mode** EMD in XML format, Calculation

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies the beginning time at which telemetry was received during a time period of interest. This should be represented in UTC system format.	**Type** datetime **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:FirstPartErt" **Type** n/a
Ops Keyword EARTH_RECEIVED_STOP_TIME **Definition** Specifies the ending time for receiving telemetry during a time period of interest. This should be represented in UTC system format.	**Valid Values** YYYY-DOYThh:mm:ss[.fff] **Type** datetime **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format, Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata: GroundCreationTime" **Type** n/a
Ops Keyword EXPECTED_PACKETS **Definition** Specifies the total number of telemetry packets which constitute a complete data product, i.e., a data product without missing data. For MSL, "Packets" are also referred to as "Parts".	**Valid Values** n/a **Type** integer **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:PartList: TotalExpected" **Type** n/a
Ops Keyword EXPECTED_TRANSMISION_PATH	**Valid Values** n/a	**Mode** EMD in XML format

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
PDS Keyword MSL:EXPECTED_TRANSMISION_PATH **Definition** Routing control at time of MPDU (Metadata PDU) generation. Indicates the planned transmission paths (routes) for the Data Product. See also COMMUNICATION_SESSION_ID.	**Type** string **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata: TransmissionControlCriterion" **Type** n/a
Ops Keyword FILE_RECORDS **Definition** Specifies the number of physical file records, including both label records and data records. NOTE: In the PDS the use of FILE_RECORDS along with other file-related data elements is fully described in the Standards Reference.	**Valid Values** 1 **Type** integer **Units** n/a **Location** FILE_DATA_ELEMENT (Class)	**Mode** Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** **Type** U16
Ops Keyword FLIGHT_SOFTWARE_MODE **PDS Keyword** MSL:FLIGHT_SOFTWARE_MODE **Definition** Active Flight Software mode at Data Product creation.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:FswMode" **Type** unsigned integer
Ops Keyword FLIGHT_SOFTWARE_VERSION_ID **Definition** The version of the instrument flight software used to acquire	**Valid Values** n/a **Type** string	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:FswVersion"

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
the image.	**Units** n/a **Location** TELEMETRY (Class)	**Type** unsigned integer
Ops Keyword ^ENGINEERING_TABLE **Definition** Specifies a pointer to the Engineering Table object. See chapter 14 of the PDS Standards Reference for more information on pointer usage.	**Valid Values** n/a **Type** NULL **Units** n/a **Location** POINTERS	**Mode** Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword ^HEADER_TABLE **Definition** Specifies a pointer to the Header Table object. See chapter 14 of the PDS Standards Reference for more information on pointer usage.	**Valid Values** n/a **Type** NULL **Units** n/a **Location** POINTERS	**Mode** Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword INSTRUMENT_HOST_ID **Definition** Specifies a unique identifier for the host where an instrument is	**Valid Values** SCID VALUE 158 "SIM" 76 "MSL"	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:Scid"

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
located. This host can be either a spacecraft or an earth base (e.g., and observatory or laboratory on the earth). Thus, INSTRUMENT_HOST_ID can contain values which are either SPACECRAFT_ID values or EARTH_BASE_ID values.	**Type** string array **Units** n/a **Location** IDENTIFICATION (Class)	**Type** n/a
Ops Keyword INSTRUMENT_HOST_NAME **Definition** 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.	**Valid Values** SCID VALUE 158 "SIMULATED MARS SCIENCE LABORATORY" 76 "MARS SCIENCE LABORATORY" **Type** string array **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:Scid" **Type** n/a
Ops Keyword INSTRUMENT_ID **Definition** 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).	**Valid Values** APXS **Type** string array **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:ProductName" **Type** n/a
Ops Keyword INSTRUMENT_NAME **Definition**	**Valid Values** "ALPHA PARTICLE X-RAY SPECTROMETER **Type**	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"**

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Specifies the full name of an instrument. Note that the associated INSTRUMENT_ID element provides an abbreviated name or acronym for the instrument. Example values: FLUXGATE MAGNETOMETER, NEAR_INFRARED MAPPING SPECTROMETER.	string array **Units** n/a **Location** IDENTIFICATION (Class)	• "MslEarthProductMetadata:MslProductMetadata:ProductName" **Type** n/a
Ops Keyword INSTRUMENT_TYPE **Definition** 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.	**Valid Values** Value "SPECTROMETER" **Type** string(15) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** **Field as "<xml name>:[<element>]:[<element>]:<field>"** **Type** n/a
Ops Keyword LOCAL_MEAN_SOLAR_TIME **PDS Keyword** MSL:LOCAL_MEAN_SOLAR_TIME **Definition** Specifies the local mean solar time, or LMST. It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. The desire to work with solar days, hours, minutes, and seconds of uniform length led to the concept of the fictitious mean Sun or FMS. The FMS is defined as a point that moves on the celestial equator of a planetary body at a constant rate that represents the average mean motion of the Sun over a planetary year. Local mean solar time is defined, by analogy with local true solar time (LTST), as the difference between the areocentric right ascensions of a point on the surface and of the FMS. The difference between LTST and LMST varies over time. The length of a mean solar day is constant and can be computed	**Valid Values** Sol-<nnnnn>M<hh>:<mm>:<ss>[.fff] NOTE: Value will be uncalibrated if SPICE kernels are unavailable. **Type** string(12) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** • Calculation: - SCLK Kernel - Landing Site Kernel - P Kernel **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:DvtCourse" • "MslEarthProductMetadata:MslProductMetadata:DvtFine" **Type** U32

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
from the mean motion of the FMS and the rotation rate of a planet. The mean solar day is also called a 'sol'. Mean solar hours, minutes, and seconds are defined in the same way as the true solar units. For MSL, the valid value is expressed in terms of a 24-hour clock, so the acceptable range is “00:00:00.000” to “23:59:59.999”. See also LOCAL_TRUE_SOLAR_TIME.		
Ops Keyword LOCAL_TRUE_SOLAR_TIME **Definition** Specifies the local true solar time, or LTST. It is one of two types of solar time used to 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	**Valid Values** hh:mm:ss[.fff] NOTE: Value will be uncalibrated if SPICE kernels unavailable. **Type** string(12) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** • Calculation: - SCLK Kernel - Landing Site Kernel - P Kernel **Field as “<xml name>:[<element>]:[<element>]:<field>”** • “MslEarthProductMetadata:MslProductMetadata:DvtCourse” • “MslEarthProductMetadata:MslProductMetadata:DvtFine” **Type** U32

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
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, 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. For MSL, the valid value is expressed in terms of a 24-hour clock, so the acceptable range is “00:00:00.000” to “23:59:59.999”. See also LOCAL_MEAN_SOLAR_TIME.		
Ops Keyword MISSION_NAME **Definition** 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.	**Valid Values** “MARS SCIENCE LABORATORY” **Type** string array **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** Static Value **Field as “<xml name>:[<element>]:[<element>]:<field>”** n/a **Type** n/a
Ops Keyword	**Valid Values**	**Mode**

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
MISSION_PHASE_NAME **Definition** Specifies the commonly-used identifier of a mission phase.	"CRUISE", "PRIMARY MISSION", "ASSEMBLY TEST LAUNCH AND OPS 1", "ASSEMBLY TEST LAUNCH AND OPS 2", "SURFACE OPS READINESS TEST 1", "SURFACE OPS READINESS TEST 2", "SURFACE OPS READINESS TEST 3", "SURFACE OPS READINESS TEST 4" **Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	User specified paramter value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword OBSERVATION_ID **Definition** Specifies a unique identifier for a scientific observation within a data set. It is set via the data product context ID - which doesn't necessarily map to a specific object - it's just used to group various instrument data sets together via a common keyword.	**Valid Values** n/a **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** DPO in XML format (referenced to APID Name in Appendix C) **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** **n/a**
Ops Keyword OBSERVATION_TYPE **Definition** Specifies the type of scientific observation within a data set. It is set via the data product context ID - which doesn't necessarily map to a specific object - it's just used to group various instrument data sets together via a common keyword.	**Valid Values** n/a **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** DPO in XML format (referenced to APID Name in Appendix C) **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:SequenceId" **Type** n/a
Ops Keyword	**Valid Values**	**Mode**

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
ORIGIN_OFFSET_VECTOR **Definition** 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 MSL, here is an example: In the case of the RSM_COORDINATE_SYSTEM group, ORIGIN_OFFSET_VECTOR describes the rotation of the RSM (camera head) boresight (about the ORIGIN_OFFSET_VECTOR) relative tothe Rover frame.	n/a **Type** float array[3] **Units** meters **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:RoverPosition" **Type** F32[3]
Ops Keyword ORIGIN_ROTATION_QUATERNION **Definition** 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 rotation occurs For all the above, the Quaternion is received in the order: (v1, v2, v3, s) For MSL, the value for ORIGIN_ROTATION_QUATERNION that defines a coordinate frame like Rover frame is computed with respect to only the orientations of the frame's axes… regardless of whether POSITIVE_ELEVATION_DIRECTION is	**Valid Values** n/a **Type** float array[4] **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** 1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:RoverAttitude" **Type** F32[4]

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
declared to be “UP” or “DOWN”. For MSL, here is an example: In the case of the RSM_COORDINATE_SYSTEM group, ORIGIN_ROTATION_QUATERNION describes the rotation of the RSM (camera head on Mast) boresight (about the ORIGIN_OFFSET_VECTOR) relative to the Rover frame.		
Ops Keyword PDS_VERSION_ID **Definition** 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.	**Valid Values** “PDS<*version*>” **Type** string[6] **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** PDS **Field as “<xml name>:[<element>]:[<element>]:<field>”** n/a **Type** n/a
Ops Keyword PLANET_DAY_NUMBER **Definition** Specifies the number of sidereal days (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 MSL, the reference day is “1”, as Landing day is Sol 1. If before Landing day, then value will be less than “1” and can be negetive.	**Valid Values** Mission Phase — Values Cruise — less than 1 Surface — “1” to n NOTE: Value will be uncalibrated if SPICE kernels are unavailable. **Type** integer **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** • Calculation: - SCLK Kernel **Field as “<xml name>:[<element>]:[<element>]:<field>”** • “MslEarthProductMetadata:MslProductMetadata:DvtCourse” • “MslEarthProductMetadata:MslProductMetadata:DvtFine” **Type** U32
Ops Keyword POSITIVE_AZIMUTH_DIRECTION	**Valid Values** “CLOCKWISE” for both START and END	**Mode** Static Value:

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition 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. For MSL, an example is, if a MastCam image is taken of the sky at an elevation 45 degrees above the horizon, the elevation coordinate in MAST_FRAME would be +0.785398 radians.	**Type** string **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	• determined by Coordinate Frame definitions **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword POSITIVE_ELEVATION_DIRECTION **Definition** 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 MSL operational coordinate frames, which follow the Mars Pathfinder convention, increasing elevation ("UP") moves towards the negative Z axis.	**Valid Values** "UP" for both START and END **Type** string **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** Static Value: • determined by Coordinate Frame definitions **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword PROCESSING_HISTORY_TEXT **Definition** Specifies an entry for each processing step and program used in generating a particular data file.	**Valid Values** "CODMAC LEVEL 1 TO LEVEL 2 CONVERSION VIA JPL/MIPL MSLEDRGEN" **Type** string **Units** n/a **Location** HISTORY (Class)	**Mode** Static Value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Ops Keyword PRODUCER_INSTITUTION_NAME **Definition** Specifies the identity of a university, 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.	**Valid Values** “MULTIMISSION INSTRUMENT PROCESSING LABORATORY, JET PROPULSION LAB” **Type** string(60) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** Static Value **Field as “<xml name>:[<element>]:[<element>]:<field>”** n/a **Type** n/a
Ops Keyword PRODUCT_COMPLETION_STATUS **Definition** Status of the product at the time that ground software wrote the data product.	**Valid Values** “PARTIAL”, “COMPLETE_NO_CHECKSUM”, ”COMPLETE_CHECKSUM_PASS”, ”PARTIAL_CHECKSUM_FAIL” **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** “MslEarthProductMetadata:MslProductMetadata:GroundStatus” **Type** n/a
Ops Keyword PRODUCT_CREATION_TIME **Definition** Specifies the UTC system format for the time when a product was created.	**Valid Values** YYYY-DOYThh:mm:ss[.fff] **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** Calculation **Field as “<xml name>:[<element>]:[<element>]:<field>”** n/a **Type** n/a
Ops Keyword PRODUCT_ID	**Valid Values** n/a	**Mode** Filename minus extension

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies a permanent, unique identifier assigned to a data product by its producer. For MSL, it is the filename minus the extension. NOTES: In the PDS, the value assigned to product_id must be unique within its data set. The PRODUCT_ID can describe the lowest-level data object that has a PDS label.	**Type** string(40) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword PRODUCT_TAG **Definition** Data Product Tag. Comment: Use of this tag is defined separately for individual product types. It is anticipated that this tag may be used to associate multiple products for later processing; it may also be used to indicate instrument FSW versions, or other uses.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:ProductTag" **Type** n/a
Ops Keyword PRODUCT_TYPE **Definition** Identifies the type or category of a data product within a data set.	**Valid Values** APXS EDR **Type** string(40) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** Static Value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword PRODUCT_VERSION_ID	**Valid Values** i.e. "V1.0"	**Mode** User specified parameter value

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition 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 MER, 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.	**Type** string(12) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword RECEIVED_PACKETS **Definition** Specifies the total number of telemetry packets which constitute a reconstructed data product.	**Valid Values** n/a **Type** integer **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:PartList: TotalReceived" **Type** n/a
Ops Keyword RECORD_BYTES **Definition** Specifies the number of bytes in a physical file record, including record terminators and separators. NOTE: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.	**Valid Values** "0" to n **Type** integer **Units** n/a **Location** FILE (Class)	**Mode** Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword RECORD_TYPE	**Valid Values** "FIXED_LENGTH"	**Mode** Calculation

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition 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.	**Type** string(20) **Units** n/a **Location** FILE (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword REFERENCE_COORD_SYSTEM_INDEX **Definition** 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. For MSL, these indices are based on the ROVER_MOTION_COUNTER and are in the same order as specified by ROVER_MOTION_COUNTER_NAME. The number of indices can be anything from 1 (used for SITE_FRAME) up to 10; however only 1, 2, 3, and 10 indices are common in RDRs. EDRs will contain 1 or 10, depending on the group in which the keyword occurs. See also REFERENCE_COORD_SYSTEM_NAME and COORDINATE_SYSTEM_INDEX.	**Valid Values** n/a **Type** integer array[1] **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** 1) EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. 2) EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:RoverMotionCounter" **Type** U16
Ops Keyword REFERENCE_COORD_SYSTEM_NAME **Definition** Specifies the full name of the reference coordinate system (CS) for the group in which the keyword occurs. All vectors and positions relating to 3-D space within the enclosing group are expressed using this reference coordinate system. For rover or lander missions with non-unique coordinate systems (such as "SITE"), the CS name and index (see (REFERENCE_COORD_SYSTEM_INDEX) together, along with the solution id (see	**Valid Values** "SITE_FRAME" for both START and END **Type** string(20) **Units** n/a **Location** 1) START_ROVER_COORDINATE_SYSTEM (Group) 2) ROVER_COORDINATE_SYSTEM (Group)	**Mode** Static Value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
REFERENCE_COORD_SYSTEM_SOLN_ID), completely specify the reference CS. The values appearing in a reference index depend on what the reference name is. The valid coordinate frame names vary by project, but are generally the same as for COORDINATE_SYSTEM_FRAME. For MSL, EDR's use a standard, predefined frame name for each occurrence. However, RDR's can use any value available in COORDINATE_SYSTEM_NAME. Despite that, only a few frame names are commonly used. “SITE_FRAME” is used for most SITE, ROVER, and LOCAL_LEVEL CS definitions, as well as for XYZ data and many mosaics. “ROVER_FRAME” is used for most other CS definitions, surface normals, camera models, and some mosaics. “LOCAL_LEVEL_FRAME” is used for some mosaics.		
Ops Keyword RELEASE_ID **Definition** 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 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 MSL the first release of the SSI EDR data set on MSL will have RELEASE_ID = "0001". The next SSI EDR release will have RELEASE_ID = "0002".	**Valid Values** n/a **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** User parameter input **Field as “<xml name>:[<element>]:[<element>]:<field>”** n/a **Type** n/a
Ops Keyword REQUEST_ID **PDS Keyword** MSL:REQUEST_ID **Definition** Specifies the Request ID value associated with the Data Product generation command. Unsigned integer.	**Valid Values** n/a **Type** string **Units** n/a **Location**	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** “MslEarthProductMetadata:MslProductMetadata:RequestId” **Type** U16

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
	IDENTIFICATION (Class)	
Ops Keyword ROVER_MOTION_COUNTER **Definition** Specifies a set of integers which describe a (potentially) unique location (position/orientation) for a rover. Each time something happens that moves, or could potentially move, the rover, 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 ROVER frames. The motion counter is defined in a mission-specific manner. Although the original intent was to have incrementing indices (e.g. MSL), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values. For MSL, the motion counter consists of ten values. In order, they are “Site”, “Drive”, “Pose”, “Arm”, “CHIMRA”, “Drill”, “RSM”, “HGA”, “DRT”, and “Inlet Cover Motion”. The Site value increments whenever a new major Site frame is declared. The Drive value increments any time intentional driving is done. Each of those resets all later indices to 0 when they increment. The Arm, RSM, and HGA increment whenever the corresponding articulation device moves. Arm, RSM and HGA increment independently of each other; they are reset to zero only when the SITE or DRIVE changes.	**Valid Values** n/a **Type** integer array[10] **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** “MslEarthProductMetadata:MslProductMetadata:RoverMotionCounter” **Type** U16
Ops Keyword ROVER_MOTION_COUNTER_NAME	**Valid Values** ("SITE", “DRIVE”, “POSE”, “ARM”, “CHIMRA”, “DRILL”, “RSM”,	**Mode** Static Values

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies an array that provides the formal names identifying each integer in ROVER_MOTION_COUNTER.	"HGA", "DRT", "IC") **Type** string array[10] **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** **Type** U16
Ops Keyword ^SCIENCE_TABLE **Definition** Specifies a pointer to the Science Table object. See chapter 14 of the PDS Standards Reference for more information on pointer usage.	**Valid Values** n/a **Type** NULL **Units** n/a **Location** POINTERS	**Mode** Calculation **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword SEQUENCE_EXECUTION_COUNT **Definition** Set to 0 at RCE start-up and Incremented each time this sequence has executed since last RCE start-up. Unsigned integer.	**Valid Values** n/a **Type** integer **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata: SequenceExecutionCounter" **Type** n/a
Ops Keyword SEQUENCE_ID	**Valid Values** "UNK"	**Mode** EMD in XML format

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies an identification of the spacecraft sequence associated with the given product. This element replaces the older seq_id, which should no longer be used.	**Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:SequenceId" **Type** n/a
Ops Keyword SEQUENCE_VERSION_ID **Definition** Specifies the version identifier for a particular observation sequence used during planning or data processing.	**Valid Values** "UNK" **Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:SequenceVersion" **Type** n/a
Ops Keyword SOFTWARE_NAME **Definition** Specifies the name of data processing software such as a program or a program library.	**Valid Values** "MSLEDRGEN", other **Type** string(60) **Units** n/a **Location** HISTORY (Class)	**Mode** Static Value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword SOFTWARE_VERSION_ID	**Valid Values** n/a	**Mode** Static Value

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies the version (development level) of a program or a program library.	**Type** string(20) **Units** n/a **Location** HISTORY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword SOLAR_LONGITUDE **Definition** 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. Example: For IRAS, the geocentric ecliptic longitude (B1950) of the Sun at the start of a scan.	**Valid Values** "0.0" to "359.99" "N/A" if any SPICE kernel is unavailable. **Type** float **Units** deg (<deg> unit tag required) **Location** IDENTIFICATION (Class)	**Mode** • Calculation: - SCLK Kernel - Landing Site Kernel - P Kernel **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:DvtCourse" • "MslEarthProductMetadata:MslProductMetadata:DvtFine" **Type** U32
Ops Keyword SPACECRAFT_CLOCK_CNT_PARTITION **Definition** Specifies the clock partition active for the SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT elements.	**Valid Values** "1" **Type** integer **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** Static Value **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword SPACECRAFT_CLOCK_START_COUNT	**Valid Values** sssssssss.mmm	**Mode** EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId.

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies the value of the spacecraft clock at the beginning of a time period of interest. Format is “sssssssss.mmm”, stored as a floating point number where, “sssssssss” = seconds converted from clock’s coarse counter “mmm” = milliseconds converted from clock’s fine counter For MSL, the fractional component “mmm” is computed as: [DvtFine/ 65536] * 1000	**Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as “<xml name>:[<element>]:[<element>]:<field>”** • “MslEarthProductMetadata:MslProductMetadata:DvtCoarse” • “MslEarthProductMetadata:MslProductMetadata:DvtFine” NOTES: • The DvtFine is a 16-bit truncated value from a 20-bit number. **Type** U32 U16
Ops Keyword SPACECRAFT_CLOCK_STOP_COUNT **Definition** Specifies the value of the spacecraft clock at the end of a time period of interest. Format is “sssssssss.mmm”, stored as a floating point number where, “sssssssss” = seconds converted from clock’s coarse counter “mmm” = milliseconds converted from clock’s fine counter For MSL, the fractional component “mmm” is computed as: [DvtFine/65536] * 1000	**Valid Values** sssssssss.mmm **Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as “<xml name>:[<element>]:[<element>]:<field>”** • “MslEarthProductMetadata:MslProductMetadata:DvtCoarse” • “MslEarthProductMetadata:MslProductMetadata:DvtFine” NOTES: • The DvtFine is a 16-bit truncated value from a 20-bit number. **Type** U32 U16
Ops Keyword SPICE_FILE_NAME	**Valid Values** n/a	**Mode** User parameter input

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition 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.	**Type** string(180) **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword START_TIME **Definition** 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.	**Valid Values** YYYY-DOYThh:mm:ss[.fff] NOTE: Value will be uncalibrated if SPICE kernels unavailable. **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format from the ApxsStart data product where groupId matches ApxsCwa or ApxsScienceAndEng groupId. **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:DvtCoarse" • "MslEarthProductMetadata:MslProductMetadata:DvtFine" NOTES: • The DvtFine is a 16-bit truncated value from a 20-bit number. **Type** U32 U16
Ops Keyword STOP_TIME **Definition** 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.	**Valid Values** YYYY-DOYThh:mm:ss[.fff] NOTE: Value will be uncalibrated if SPICE kernels unavailable. **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** • "MslEarthProductMetadata:MslProductMetadata:DvtCoarse" • "MslEarthProductMetadata:MslProductMetadata:DvtFine" NOTES: • The DvtFine is a 16-bit truncated value from a 20-bit number. **Type** U32 U16
Ops Keyword TARGET_NAME	**Valid Values** "MARS"	**Mode** Calculation: • "MARS"

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE.	**Type** string(30) **Units** n/a **Location** IDENTIFICATION (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword TARGET_TYPE **Definition** Specifies the type of a named target.	**Valid Values** "PLANET" **Type** string **Units** n/a **Location** IDENTIFICATION (Class)	**Mode** "PLANET" in database **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword TELEMETRY_PROVIDER_ID **Definition** Specifies the provider and version of the telemetry data used in the generation of this data.	**Valid Values** "MPCS_MSL_DP" **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** User parameter input **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword TELEMETRY_SOURCE_CHECKSUM	**Valid Values** n/a	**Mode** EMD in XML format

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition File checksum is an unsigned add of each byte in the data areas of the DPOs in the product. This does not include the DPO headers.	**Type** string **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:SessionInformation:ProductChecksum" **Type** n/a
Ops Keyword TELEMETRY_SOURCE_HOST_NAME **Definition** Specifies the name of the host venue that provides the telemetry source used in creation of this data set. For MSL, example is "mslmstbgds1". See also TELEMETRY_SOURCE_NAME.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:SessionInformation:Venue:Host" **Type** n/a
Ops Keyword TELEMETRY_SOURCE_NAME **Definition** Specifies the name of the telemetry source used in creation of this data set. See also TELEMETRY_SOURCE_HOST_NAME.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:DataFileName" **Type** n/a
Ops Keyword TELEMETRY_SOURCE_SIZE	**Valid Values** "DATA PRODUCT"	**Mode** User parameter input

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition Specifies the length of the Data Product in bytes. This value is *not* a "less one" value; that is, the field value is the exact length of the user data within the data product.	**Type** integer **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:ProductFileSize" **Type** n/a
Ops Keyword TELEMETRY_SOURCE_TYPE **Definition** Specifies the classification of the source of the telemetry used in creating this data set.	**Valid Values** "DATA PRODUCT" **Type** string(12) **Units** n/a **Location** TELEMETRY (Class)	**Mode** User parameter input **Field as "<xml name>:[<element>]:[<element>]:<field>"** n/a **Type** n/a
Ops Keyword TRANSMISSION_PATH **Definition** Routing status at time of MPDU (Metadata PDU) generation. Indicates the actual transmission paths (routes) of the Data Product.	**Valid Values** n/a **Type** string **Units** n/a **Location** TELEMETRY (Class)	**Mode** EMD in XML format **Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata: TransmissionStatus" **Type** n/a
Ops Keyword VIRTUAL_CHANNEL_ID	**Valid Values** n/a	**Mode** EMD in XML format

OUTPUT METADATA (PRODUCT LABEL)		INPUT METADATA (SOURCE)
• Ops Keyword • PDS-Compliant Keyword (listed if different than Ops) • Definition	• Valid Values (quoted) • Type • Units • Keyword Location in Label	• Mode • Metadata Field • Type
Definition The Virtual Channel Identifier is used by MSL to identify the RCE string generating the Transfer Frame, and to indicate the type of data flowing in the telemetry virtual channel. RCE String A is indicated by all Virtual Channel Identifier values having a '0' as the high bit (e.g., virtual channels 0 to 31); RCE String B is indicated by all Virtual Channel Identifier values having a '1' for the high bit (e.g., virtual channels 32 to 63).	**Type** integer **Units** n/a **Location** TELEMETRY (Class)	**Field as "<xml name>:[<element>]:[<element>]:<field>"** "MslEarthProductMetadata:MslProductMetadata:Vcid" **Type** n/a