

# Mars Exploration Rover (MER)

## Software Interface Specification

Interface Title: **APXS Experiment Data Record (EDR)**

Mission: MER Date: July 30, 2004

Module ID: SIS-SCI008-MER

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

Reference Module ID: N/A

Date: N/A

### Signatures

**GDS Generating Elements:**

Ops Product Generation Subsystem (OPGS)  
Justin Maki

\_\_\_\_\_  
Subsystem Engineer Date

**GDS Receiving Elements:**

MER GDS SOAS Subsystem Engineer  
Deborah Bass

\_\_\_\_\_  
Subsystem Engineer Date

**Concurrence:**

APXS Payload Element Lead  
Rudolf Rieder

\_\_\_\_\_  
PEL Date

MER APXS Investigation Scientist  
Albert Yen

\_\_\_\_\_  
Investigation Scientist Date

MER Science Manager  
John Callas

\_\_\_\_\_  
Manager Date

PDS Central Node Data Engineer  
Betty Sword

\_\_\_\_\_  
Data Manager Date

MER MIPL Cognizant Engineer  
Doug Alexander

\_\_\_\_\_  
Cognizant Engineer Date

**GDS:**

Ground Data System (GDS)  
Frank Singleton

\_\_\_\_\_  
GDS SE Date

# **Mars Exploration Rover (MER) Project**

## **APXS EDR Software Interface Specification (SIS)**

### **Version 2.01**

Prepared by:  
Edward A. Guinness  
MER APXS SIS Custodian

Helen Mortensen

Paper copies of this document may not be current and should not be relied on for official purposes. The current version is in the MER Project Library at <http://mars03-lib.jpl.nasa.gov>, in the Controlled Documents and Records folder.

JPL D-22848

July 30, 2004



Jet Propulsion Laboratory  
California Institute of Technology

## CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
9/15/01	All	First draft	Draft v0.1
11/18/01	All	MIPL additions	Draft v0.2
1/16/02	All	Second draft	Draft v0.3
7/31/02	All	Work off TBD items	Version 1
10/12/02	Sec. 2.3.3, Sec. 2.3.4, Appendix A, Appendix B	Update to reflect decision to not support multiple instances of groups, support unique group names, use a Data Product as the input to mertelemproc, and update the file naming convention. Deleted the TELEMETRY_PROVIDER_TYPE, SOURCE_ID, and GROUP_ID keywords.	Version 1.1
11/26/02	Cover Page, 2.3.2, 3.2, Appendix A	Replace Craig Leff with Arthur Amador, update to reflect data products, update to add new temperature correction factors and engineering data component.	Version 1.2
12/3/02	3.2 Table 4, Table 5	Added additional fields according the APXS IICD document	Version 1.2
12/5/02	3.2 Table 4, Table 5, appendix A	Update with information from Ralf Gellert on the temperature compensation. Change Alpha spectra references to alpha1 and changed background to alpha2 in the text and in the label.	Version 1.2
12/10/02	1.3, 2.1, 2.2, 2.3.4, 3.2 Table 4, Table 5, Appendix A	Added new reference. Update timing information. Update with more detailed information from Ralf Gellert and Ed Guinness on engineering data and change from background to alpha2. Change order of temperature data.	Version 1.2
12/12/02	Table 4, Table 5, Appendix A	Changed ADDRESS to ADDRESS and made all integers UNSIGNED.	Version 1.2
12/20/02	Update 1.3 and Appendix A	Update references and change BEGIN to START and END to STOP in group names.	Version 1.2
3/7/03	All	Removed SOAS as receiving element. Updated the file naming convention, removed PRODUCER_ID and INSTRUMENT_VERSION_ID. Several updates to the example label. Added RELEASE_ID	Version 2.0
3/28/03	2.3.4, Appendix A & B,	File naming convention update. Add Z to time formats, PRODUCT_VERISON_ID update.	Version 2.0

6/13/03	2.3.4, Appendix A & B	File naming convention update. Updated label and definitions. Removed SPICE_FILE_ID	Version 2.0
9/15/03	TBD, 2.2, 3.2, 4.2, Appendix A & B	Updates on spectrum id from Ed Guinness. Added LOCAL_TRUE_SOLAR_TIME to label. Removed section 4.2.	Version 2.0
10/7/03	Appendix A & B	Removed TLM_CMD_DISCREPANCY_FLAG keyword. Added SAMPLING_COUNT to address TBD item for needing TELEMETRY_COUNT keyword.	Version 2.0
11/6/03	Acronyms, Figure 1, Appendix A	Update coordinate system diagram, fix typos.	Version 2.0
7/30/04	Appendix A	<p>For archive data, added double quotes to  RELEASE_ID,  LOCAL_TRUE_SOLAR_TIME,  OBSERVATION_ID,  SEQUENCE_VERSION_ID,  SPACECRAFT_CLOCK_START_COUNT,  SPACECRAFT_CLOCK_STOP_COUNT.</p> <p>Removed double quotes from  PRODUCT_CREATION_TIME,  START_TIME, STOP_TIME,  EARTH_RECEIVED_START_TIME,  EARTH_RECEIVED_STOP_TIME.</p> <p>Added OPS to DATA_SET_ID.</p>	VERSION 2.01

## TBD ITEMS

SECTION	DESCRIPTION
---------	-------------

## CONTENTS

CHANGE LOG .....	II
TBD ITEMS .....	IV
CONTENTS .....	V
LIST OF FIGURES .....	VI
LIST OF TABLES .....	VI
ACRONYMS AND ABBREVIATIONS .....	VII
GLOSSARY .....	IX
1. INTRODUCTION .....	1
1.1 Purpose and Scope .....	1
1.2 Contents .....	1
1.3 Applicable Documents and Constraints .....	1
1.4 Relationships with Other Interfaces .....	2
2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT .....	2
2.1 Instrument Overview .....	2
2.2 Data Product Overview .....	3
2.3 Data Processing .....	3
2.3.1 Data Processing Level .....	3
2.3.2 Data Product Generation .....	4
2.3.3 Data Flow .....	4
2.3.4 Labeling and Identification .....	5
2.4 Standards Used in Generating Data Products .....	7
2.4.1 PDS Standards .....	7
2.4.2 Time Standards .....	7
2.4.3 Coordinate Systems .....	7
2.4.4 Data Storage Conventions .....	9
2.5 Data Validation .....	9
3. DETAILED DATA PRODUCT SPECIFICATIONS .....	10
3.1 Data Product Structure and Organization .....	10
3.2 Data Format Descriptions .....	10
3.3 Label and Header Descriptions .....	14
3.3.1 PDS Label .....	14
3.3.2 PDS Data Objects .....	14
4. APPLICABLE SOFTWARE .....	15
4.1 Utility Programs .....	15
APPENDIX A - EXAMPLE OF AN APXS-EDR LABEL .....	16
APPENDIX B – APXS LABEL KEYWORD DEFINITIONS .....	26

## LIST OF FIGURES

Figure 1: S, S <sub>R</sub> , and R Frame Coordinate Systems.....	9
Figure 2: The APXS EDR consists of two files. ....	10
Figure 3: Schematic of an APXS EDR data file.....	11
Figure 4: Data structure for an APXS measurement. ....	11

## LIST OF TABLES

Table 1: Product and Software Interfaces to this SIS .....	2
Table 2: Processing Levels for Science Data Sets.....	3
Table 3: Coordinate Frames Used for MER Surface Operations .....	8
Table 4: MER APXS Measurement Data Components .....	12
Table 5: MER APXS Engineering Data Component.....	13

## 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
EDR	Experiment Data Record
FEI	File Exchange Interface
ICD	Interface Control Document
IDD	Instrument Deployment Device
ISO	International Standards Organization
JPL	Jet Propulsion Laboratory
Kbyte	Kilobytes
LSB	Least Significant Byte
MB	Mega Bytes
MER	Mars Exploration Rover
MIPL	Multimission Image Processing Laboratory
MSB	Most Significant Byte
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
OPGS	Operations Product Generation Subsystem
OSS	Operations Storage Server
PDS	Planetary Data System
PEL	Payload Element Lead
PPPCS	Pointing, Positioning, Phasing & Coordinate Systems
RAM	Random Access Memory
RDR	Reduced Data Record
RSVP	Rover Sequence and Visualization Program
SAP	Science Activity Planner
SCM	Spacecraft Configuration Manager
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SOAS	Science Operations Analysis Subsystem
SSW	System Software
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 twelve measurements, each of which consists of an alpha, x-ray, and background spectrum, along with temperature 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 the EDR products and scientists who will analyze the data, including those associated with the Mars Exploration Rover (MER) Project and those in the general planetary science community.

## 1.2 Contents

This data product SIS describes how the MER 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 MER documents:

1. Mars Exploration Program Data Management Plan, R. E. Arvidson, S. Slavney and S. Nelson, Rev. 3, March 20, 2002.
2. Mars Exploration Rover Project Archive Generation, Validation and Transfer Plan, R. E. Arvidson and S. Slavney, JPL D-19658, March 22, 2002.
3. MER Flight-Mission Systems ICD (FMICD), Vol. 4 Command Dictionary, MER 420-3-15.04, JPL D-20616.
4. MER Flight-Mission Systems ICD (FMICD), Vol. 7 Telemetry Dictionary, MER 420-3-15.047 JPL D-20617.
5. Pointing, Positioning, Phasing & Coordinate Systems Master (PPPCS), S.R. Doudrick, JPL D-19720, June 28, 2001.

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

6. Planetary Data System Data Preparation Workbook, Version 3.1, JPL D-7669, Part 1, February 1, 1995.

7. Planetary Data System Data Standards Reference, Version 3.6, JPL D-7669, Part 2, August 1, 2003.
8. Planetary Science Data Dictionary Document, JPL D-7116, August 28, 2002.
9. Mars Exploration Rover Alpha Particle X-Ray Spectrometer Information Interface Control Document, R. Heninger, Rev. B, JPL-D20251, September 10, 2002.

Finally, this SIS is meant to be consistent with the contract negotiated between the MER Project and the Athena 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
mertelemproc	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
apxs2asc	S	Geosciences Node

## 2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

### 2.1 Instrument Overview

The MER APXS instrument is similar to the APXS flown on the Mars Pathfinder rover, but without the proton mode. The MER APXS will determine the dominant elements that make up martian surface rocks and soils by irradiating a sample with alpha particles and x-rays from a radioactive source. It will measure the energy spectra of backscattered alpha particles and emitted x-rays. The instrument consists of a sensor head mounted on the rover's Instrument Deployment Device (IDD), and electronics mounted in the rover's Warm Electronics Box. The sensor head contains six radioactive sources and six alpha detectors and one x-ray detector. The

instrument also has two temperature sensors; one located on the sensor head and one located on the electronics board. The APXS can measure two separate onboard calibration targets for post-landing calibration.

Most APXS data collection on Mars will be done at night. APXS data are stored in the instrument's RAM, which contains twelve 2.5 Kbyte sections of memory. During a measurement session, data are accumulated and stored in a 2.5 Kbyte working area of memory within the instrument's RAM. After 90 minutes, the measurement is halted and the current measurement data are shifted to the next 2.5 Kbyte section of memory, the working area is cleared, and the measurement is restarted. Thus, the APXS memory can store up to twelve sets of measurements. After twelve measurements are collected, the oldest measurement is discarded as a new 2-hr measurement starts.

## 2.2 Data Product Overview

Each APXS EDR will consist of two files. The first file is an ASCII formatted detached PDS label. The second file is a binary data file. The APXS EDR 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. The APXS EDR contains energy spectra in units of counts and temperature values. Temperatures are derived from the two temperature sensors. There are two types of APXS spectra in each measurement: one x-ray, and two alpha spectra. The energy scale is divided into discrete channels with 256 channels for the alpha spectra and 512 channels for the x-ray spectra.

Both alpha spectra use the same detectors, but the second alpha detectors are covered, thus measuring only background events.

The instrument holds data for the twelve most recent measurements in memory. When a new measurement begins, the oldest one is discarded. Each spectrum has a unique identifier. There are also 2 Kbytes at the end of the APXS EDR data file that are transferred from the instrument's internal RAM. This 2 Kbyte section of RAM is used for instrument engineering data.

## 2.3 Data Processing

### 2.3.1 Data Processing Level

This SIS uses the Committee On Data Management And Computation (CODMAC) data level numbering system to describe the processing level of the EDR data product. APXS EDR data products are considered CODMAC "Level 2" or "Edited Data" (equivalent to NASA level 0) products. The EDR data files are generated from "Level 1" or "Raw Data", which are the telemetry packets within the project specific Standard Formatted Data Unit (SFDU) record. 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.

NASA	CODMAC	Description
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 (Multimission Image Processing Laboratory) at JPL under the OPGS using the telemetry processing software, mertelemproc. The EDR data products will be raw uncalibrated data reconstructed from telemetry data products and formatted according to this EDR SIS. Meta-data acquired from the telemetry data headers will be used to populate the PDS label. There will not be multiple versions of an APXS EDR. If telemetry packets are missing during the initial downlink from the rover memory, partial data sets will be created and the missing data will be filled with [zeroes]. The data will be reprocessed after all data are received and the original version will be overwritten.

The APXS memory is maintained by a battery, and in theory, retransmits from the APXS memory could be commanded. In this instance, a new data product will be created, because there will likely be subtle and potentially useful changes in the data when a retransmit from the APXS memory is commanded.

### **2.3.3 Data Flow**

The APXS EDR data products generated by MIPL during operations are created collectively from: a) SSW data products b) SPICE kernels, and c) a meta-data database. They are created on the OSS and then deposited into MIPL's File Exchange Interface (FEI) for electronic distribution to remote via a secure subscription protocol. After a data validation period, the APXS EDR data products are collected with other science data and written to physical media for archiving with the Planetary Data System [see reference 2].

The size of the APXS EDR data file is 0.032 MB. The APXS EDR will be generated 60 seconds after the data product for the EDR has been received by MIPL. The APXS data will be reprocessed only if packets in the original downlink are not received. Partial files are created



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

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

Example value is "AK" for value of 120..

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

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

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

Example value is "AK" for value of 120..

**seq** = (1 alpha character plus 4 integers) Sequence Number. Denotes a group of related commands used as keys for the Ops processing.

Valid values for character (position 1) in field:

"C" - Cruise	"P" - PMA instr. (Pancam, Navcam, MTES)
"D" - IDD & RAT	"R" - Rover Driving
"E" - Engineering	"S" - Submaster
"F" - Flight Software (Seq rejected)	"T" - Test
"G" - (spare)	"W" - Seq. triggered by a commun. Window
"K" - (spare)	"X" - Contingency
"M" - Master (Surface only)	"Y" - (spare)
"N" - In-Situ instr. (APXS, MB, MI)	"Z" - SCM Seq's

Valid values for integers (positions 2 thru 5) in field:

**0001** thru **4095** - Valid Sequence number, commanded by Ground

Needs "F" in character position (Camera only):

- 1000** - Commanded by NAV
- 2000** - Commanded by SAPP
- 3000** - Commanded by Fault protection
- 4000** - Commanded by EDL

Example value is "N0268".

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

- "L" - Left camera eye
- "R" - Right camera eye
- "B" - Both left and right camera eyes
- "M" - Monoscopic (one camera eye)
- "N" - Not Applicable (non-image data)

**filt** = (1 integer) Filter number, with a valid range of **0-8** (0 = "no filter" or "N/A").

**who** = (1 alpha character) Product creator indicator. Valid values are as follows, though others may be added in the future:

**"A"** - Arizona State University  
**"C"** - Cornell University  
**"F"** - USGS at Flagstaff  
**"J"** - Johannes Gutenberg Univ. (Germany)  
**"M"** - OPGS (MIPL) at JPL  
**"N"** - NASA Ames Research Center  
**"P"** - Max Planck Institute (Germany)  
**"S"** - SOAS at JPL  
**"U"** - University of Arizona  
**"V"** - SSV Team (E. De Jong) at JPL  
**"X"** - Other

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

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

Range 1 thru 9 - **"1", "2", ... "9"**

Range 10 thru 35 - **"A", "B", ... "Z"**

Example value is **"E"** for value of 14.

**ext** = (3 alpha characters) PDS product type extension.

Valid values for MER non-camera instrument products:

**"QUB"** - Mini-TES Data Cube

**"DAT"** - APXS spectra, Mössbauer spectra, RAT binary data

**"TAB"** - APXS table data, Mössbauer table data

**"LBL"** - Detached PDS labels for APXS and Mössbauer data

Example:

a) 1A123456789EDR0103N0062N0M1.DAT    Rover MER-1, APXS instrument, EDR, Site 01,  
 Position 03, Seq N0062, produced by MIPL, product  
 version 1.

## 2.4 Standards Used in Generating Data Products

### 2.4.1 PDS Standards

The APXS EDR complies 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

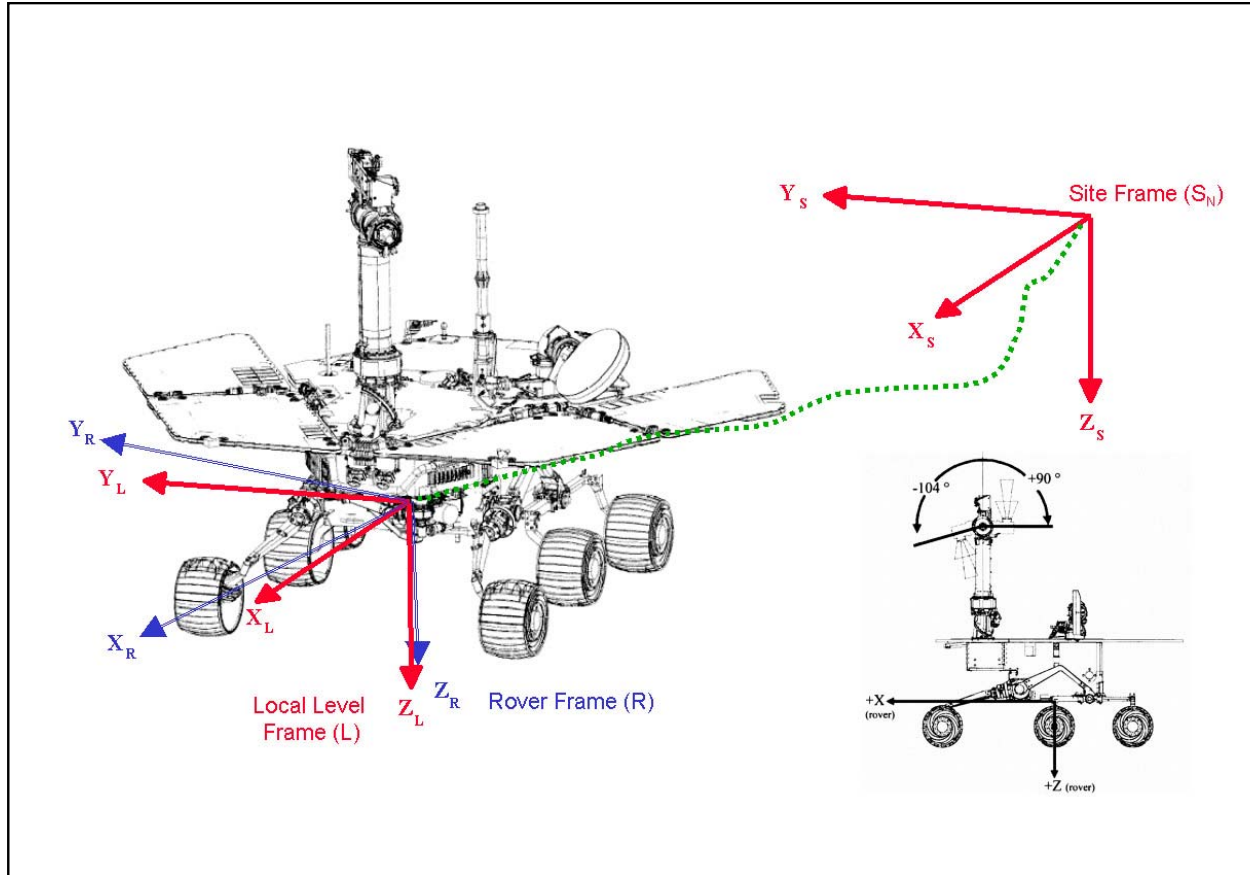
The PDS label for an APXS EDR uses keywords containing time values, such as start time, stop time, start spacecraft clock count, and stop spacecraft clock count. Each time value standard is defined according to the keyword definition. See Appendix B.

### 2.4.3 Coordinate Systems

The coordinate systems defined for MER surface operations are listed in Table 3 and illustrated in Figure 1 below. Refer to the Pointing, Positioning, Phasing and Coordinate Systems document [5] for more details. The APXS coordinate system is defined relative to the Rover XYZ coordinate system (ROVER\_FRAME).

**Table 3: Coordinate Frames Used for MER Surface Operations**

IMAGING-RELATED COORDINATE SYSTEMS		COORDINATE SYSTEM ORIGIN	COORDINATE SYSTEM ORIENTATION
NAME	LABEL KEYWORD		
Lander Frame (L Frame)	"LANDER_FRAME"	Attached to Lander	Aligned with Lander
Mars Body Fixed (MBF)	does not appear in label	Attached to Mars center of Mass	x=equatorial plane, intersects the prime meridian, z= Mars spin axis, points toward the North pole, y completes the right-handed system
Mast Frame	"MAST_FRAME"	Attached to PMA mast head	Aligned with pointing of mast head
Pancam Frame	"PANCAM_FRAME"	Attached to Camera	Aligned with camera pointing
Rover Frame (R Frame)	"ROVER_FRAME"	Attached to Rover	Aligned with Rover
Surface ( $S_n$ Frame) (Site Frame)	"SITE_FRAME"	Attached to Surface	North/East/Nadir
Surface Rover ( $S_R$ Frame) (Local Level)	"LOCAL_LEVEL_FRAME"	Attached to Rover (coincident with Rover Frame)	North/East/Nadir



**Figure 1: S, S<sub>R</sub>, and R Frame Coordinate Systems**

#### **2.4.4 Data Storage Conventions**

The APXS EDR data files contain binary data. Spectral data are 16-bit 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 MER 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

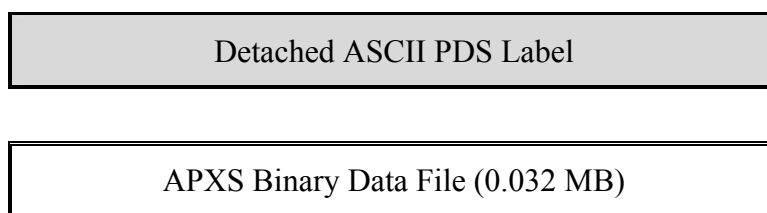
identified for automated verification, they will be added to the validation procedure.

Manual validation of the data will be performed both as 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). These products will be viewed by a human. Validation in this case will include inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the target shown / apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure.

### 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 EDR data file consist of a 32 Kbyte block of APXS binary data with a detached ASCII PDS label (see sec. 3.2.1). Each 32 Kbyte block of APXS data is a copy of the instrument's memory and contains data for twelve measurements. Each measurement consists of APXS spectra and temperature data. The APXS EDR also contains a 2 Kbyte block of engineering data from the instrument's internal RAM (Figure 3).

An APXS measurement counts events for up to 90 minutes. Data remain in APXS memory buffer until 12 new measurements are collected. Thus, an EDR contains data collected for up to 18 hours. Each spectrum is automatically given a unique identifier by the APXS instrument and this identifier is stored with the spectrum.

APXS Measurement 1 (2.5 Kbytes)
APXS Measurement 2 (2.5 Kbytes)
⋮
APXS Measurement 12 (2.5 Kbytes)
APXS Engineering Data (2 Kbytes)

**Figure 3: Schematic of an APXS EDR data file.**

Each APXS measurement is 2.5 Kbytes long and contains 1024 bytes of x-ray data, two 512 bytes of alpha particle data, and 512 bytes of temperature data (Figure 4). The detailed measurement record structure is shown in Table 4. Alpha particle spectra contain 256 channels and x-ray spectra have 512 channels. Values for all channels are 16-bit binary integers in LSB first order. The first channel (channel 0) of each spectrum contains the time that the instrument was available to count events. This length of time, termed the lifetime, is stored in units of 10 seconds. It gives the time the instrument was able to detect signals. This means, measuring time minus processing time. When a signal is processed, other signals were ignored during the period. The value of lifetime should nominally be 540 (i.e., total lifetime of 5400 seconds or 90 minutes). The twelve least significant bits of the second channel (channel 1) of each spectrum contains a sequential number generated by the instrument to identify the measurement. The value is incremented by one after every cycle start command or internal cycle. Channels 2 and 3 in all but the temperature data are the temperature correction factors that were valid when the spectra were started. Temperature compensation values A0 and G are 16-bit values, MSB first for all spectra. Channels 4 (2 for temperature) through 254 (or 510 for x-ray data) are energy channels and contain the number of events counted within a given energy interval. The last channel of each spectrum contains the number of events with energies greater than the full scale. For the 16-bit count data, the two bytes can be converted into an integer value as follows:  $\text{value} = \text{LSB} + (256 * \text{MSB})$ .

X-ray Spectra (1024 bytes)	Alpha1 Spectra (512 bytes)	Alpha2 Spectra (512 bytes)	Temperature Data (512 bytes)
-------------------------------	-------------------------------	-------------------------------	---------------------------------

**Figure 4: Data structure for an APXS measurement.**

Temperatures at the instrument sensor head and on the electronics board are recorded at a 30 second interval during a measurement. These temperature data are used for internal adjustments of temperature-sensitive instrument parameters. Temperature values are stored in pairs of two 8-

bit numbers with the first byte containing the electronics board temperature and the second byte containing the temperature of the sensor head. The 8-bit digital value can be multiplied by 1.442 to convert the data into Kelvin.

**Table 4: MER APXS Measurement Data Components**

Spectrum	Channels	Bytes	Description
X-ray	0	1-2	Lifetime in units of 10 seconds. 16-bit LSB first unsigned integer.
	1	3-4	Spectrum identifier. Incremented by one after every cycle start command or internal cycle. 12-bit LSB first unsigned integer.
	2	5-6	A0 or the multiplier for the gain. 16-bit MSB first unsigned integer. (8000 hex means 1)
	3	7-8	G or the linear term of the Temperature Compensation. 16-bit MSB first unsigned integer. (0000 hex means no TC)
	4 – 510	9-1022	Event counts; each channel is a separate energy interval. 16-bit LSB first unsigned integers.
	511	1023-1024	Overflow counts; number of events with energies greater than full scale. 16-bit LSB first unsigned integer.
Alpha1	0	1025-1026	Lifetime in units of 10 seconds. 16-bit LSB first unsigned integer.
	1	1027-1028	Spectrum identifier. Incremented by one after every cycle start command or internal cycle. 12-bit LSB first unsigned integer.
	2	1029-1030	A0 or the multiplier for the gain. 16-bit MSB first unsigned integer. (8000 hex means 1)
	3	1031-1032	G or the linear term of the Temperature Compensation. 16-bit MSB first unsigned integer. (0000 hex means no TC)
	4 – 254	1033-1534	Event counts; each channel is a separate energy interval. 16-bit LSB first unsigned integers.
	255	1535-1536	Overflow counts; number of events with energies greater than full scale. 16-bit LSB first unsigned integer.
Alpha2 (Background)	0	1537-1538	Lifetime in units of 10 seconds. 16-bit LSB first unsigned integer.
	1	1539-1540	Spectrum identifier. Incremented by one after every cycle start command or internal cycle. 12-bit LSB first unsigned integer.

Spectrum	Channels	Bytes	Description
	2	1541-1542	A0 or the multiplier for the gain. 16-bit MSB first unsigned integer. (8000 hex means 1)
	3	1543-1544	G or the linear term of the Temperature Compensation. 16-bit MSB first unsigned integer. (0000 hex means no TC)
	4 – 254	1545-2046	Event counts; each channel is a separate energy interval. 16-bit LSB first unsigned integers.
	255	2047-2048	Overflow counts; number of events with energies greater than full scale. 16-bit LSB first unsigned integer.
Temperature Data	0 - 255	2049-2560	Pairs of 8-bit unsigned integers. WEB temperature is the first byte, sensor head temperature is the second byte; measured at approximately 30 second intervals.

The engineering data consist of 2Kbytes of data and is located after the 12<sup>th</sup> measurement.

**Table 5: MER APXS Engineering Data Component**

Channels	Bytes	Description
	1- 2	A0 TC (xray) 16-bit MSB first unsigned integer
	3 - 4	G TC (xray) 16-bit MSB first unsigned integer
	5 - 6	A0 TC (alpha1) 16-bit MSB first unsigned integer
	7 - 8	G TC (alpha1) 16-bit MSB first unsigned integer
	9 – 10	A0 TC (alpha2) 16-bit MSB first unsigned integer
	11 - 12	G TC (alpha2) 16-bit MSB first unsigned integer
	13	Cycle interval/min. unsigned integer
	14	Spare byte, adjustable, not used
	15–23	Reserved bytes (9 bytes)

Channels	Bytes	Description
	24-25	Uptime from the last power up or software reset/10s. (16-bit LSB first unsigned integer)
	26-31	Reserved bytes (6 bytes)
	32-33	Address of the last command written into the logbook. (LSB + MSB * 256 – F700hex + 1) (16-bit MSB first unsigned integer address).
	34-254	Reserved bytes, (221 bytes)
	255-2048	Circular written log book of received command bytes. A power up is indicated by two 00hex. For every command received by the instrument, the command opcode byte is written into the logbook and the address of the logbook counter is increased. Unsigned integer.

### 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:

$$\wedge\text{object} = \text{location}$$

where the carat character ( $\wedge$ , 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 temperature data. It has 12 rows with each row containing an APXS measurement (Figure 3). The length of each row is 2560 bytes. The second data object in the APXS EDR data file is a table containing the spare RAM bytes from the APXS instrument. This table has one row that is 2048 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.

**APPENDIX A - EXAMPLE OF AN APXS-EDR LABEL**

```

PDS_VERSION_ID           = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE              = FIXED_LENGTH
RECORD_BYTES             = 512
FILE_RECORDS             = 64
^MEASUREMENT_TABLE      = ("1A123456789EDR0103C0062N0M1.DAT", 1)
^ENGINEERING_TABLE      = ("1A123456789EDR0103C0062N0M1.DAT", 61)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID              = "MER1-M-APXS-2-EDR-V1.0"
PRODUCT_ID               = "1A123456789EDR0103C0062N0M1"
PRODUCT_TYPE             = APXS_EDR
PRODUCT_VERSION_ID      = "V2.0 D-22848"
RELEASE_ID               = "0001"
ROVER_MOTION_COUNTER     = (2, 5, 3, 1, 2)
ROVER_MOTION_COUNTER_NAME = (SITE, DRIVE, IDD, PMA, HGA)
COMMAND_SEQUENCE_NUMBER = 33
INSTRUMENT_HOST_ID      = MER1
INSTRUMENT_HOST_NAME    = "MARS EXPLORATION ROVER 1"
INSTRUMENT_ID           = APXS
INSTRUMENT_TYPE         = SPECTROMETER
LOCAL_TRUE_SOLAR_TIME   = "09:16:03"
MAGNET_ID               = "NULL"
MISSION_NAME             = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME      = <CRUISE, "EXTENDED MISSION",
                          "PRIMARY MISSION", ATLO,
                          ORT1, ORT2, TBD>
OBSERVATION_ID          = "135609876"
PLANET_DAY_NUMBER       = 2
PRODUCER_INSTITUTION_NAME = "MULTIMISSION IMAGE PROCESSING SUBSYSTEM,
                          JET PROPULSION LAB"
PRODUCT_CREATION_TIME   = YYYY-MM-DDThh:mm:ss.fff
SEQUENCE_ID             = c0062
SEQUENCE_VERSION_ID     = "1"
SPACECRAFT_CLOCK_CNT_PARTITION = 1
SPACECRAFT_CLOCK_START_COUNT = "nnnnnnnnnn.nnn"
SPACECRAFT_CLOCK_STOP_COUNT = "nnnnnnnnnn.nnn"
START_TIME              = YYYY-MM-DDThh:mm:ss.fff
STOP_TIME               = YYYY-MM-DDThh:mm:ss.fff
TARGET_NAME             = MARS
TARGET_TYPE             = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID   = 32
APPLICATION_PROCESS_NAME = APXS
APPLICATION_PROCESS_SUBTYPE_ID = 0
EARTH_RECEIVED_START_TIME = 2004-02-14T01:19:27.453

```

```

EARTH_RECEIVED_STOP_TIME      = 2004-02-14T03:37:16.153
EXPECTED_PACKETS              = "N/A"
PACKET_MAP_MASK               = "N/A"
RECEIVED_PACKETS              = "N/A"
SAMPLING_COUNT                = 5
SPICE_FILE_NAME                = "chronos.mer"
TELEMETRY_FORMAT_ID           = SPT
TELEMETRY_PROVIDER_ID         = "SSW MER_DP"
TELEMETRY_SOURCE_NAME         = "Input DP Filename"
TELEMETRY_SOURCE_TYPE         = "DATA PRODUCT"
TLM_INST_DATA_HEADER_ID       = 3

/* HISTORY DATA ELEMENTS */

SOFTWARE_NAME                  = MERTELEMPROC
SOFTWARE_VERSION_ID            = "V1.24.46"
PROCESSING_HISTORY_TEXT        = "CODMAC LEVEL 1 TO LEVEL 2 CONVERSION
                                VIA JPL/MIPL MERTELEMPROC"

/* ROVER STATE */
/* COORDINATE SYSTEM STATE: ROVER */

GROUP                          = ROVER_COORDINATE_SYSTEM
COORDINATE_SYSTEM_NAME         = ROVER_FRAME
COORDINATE_SYSTEM_INDEX        = (2, 5, 3, 1, 2)
COORDINATE_SYSTEM_INDEX_NAME   = (SITE, DRIVE, IDD, PMA, HGA)
ORIGIN_OFFSET_VECTOR           = (0.0230152, -0.076101, 0.874005)
ORIGIN_ROTATION_QUATERNION      = (0.922297, -0.0165226, -0.0413094,
                                0.382304)
POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION   = UP
QUATERNION_MEASUREMENT_METHOD  = FINE
REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
REFERENCE_COORD_SYSTEM_INDEX    = 2
END_GROUP                       = ROVER_COORDINATE_SYSTEM

/* ARTICULATION DEVICE STATE: INSTRUMENT DEPLOYMENT DEVICE AT THE START*/

GROUP                          = START_IDD_ARTICULATION_STATE
ARTICULATION_DEVICE_ID          = IDD
ARTICULATION_DEVICE_NAME        = "INSTRUMENT DEPLOYMENT DEVICE"
ARTICULATION_DEVICE_ANGLE       = (0.0230152 <rad>, -0.076101 <rad>,
                                0.874005 <rad>, 9.4095 <rad>,
                                0.3467 <rad>, 0.922297 <rad>,
                                0.0165226 <rad>,
                                0.0413094 <rad>, 0.38230 <rad>,
                                0.456 <rad>)
ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 AZIMUTH-ENCODER",
                                "JOINT 2 ELEVATION-ENCODER",
                                "JOINT 3 ELBOW-ENCODER",
                                "JOINT 4 WRIST-ENCODER",
                                "JOINT 5 TURRET-ENCODER",
                                "JOINT 1 AZIMUTH-POTENTIOMETER",
                                "JOINT 2 ELEVATION-POTENTIOMETER",
                                "JOINT 3 ELBOW-POTENTIOMETER",
                                "JOINT 4 WRIST-POTENTIOMETER",
                                "JOINT 5 TURRET-POTENTIOMETER")

```

```

ARTICULATION_DEVICE_MODE           = "FREE SPACE"
ARTICULATION_DEVICE_TEMP           = (0.922297 <degC>, -0.0165226 <degC>)
ARTICULATION_DEVICE_TEMP_NAME      = ("AZIMUTH JOINT 1", "TURRET JOINT 5")
ARTICULATION_DEV_VECTOR            = (1.23456, 3.4567, 23.456)
ARTICULATION_DEV_VECTOR_NAME       = GRAVITY
CONTACT_SENSOR_STATE               = ("NO CONTACT", "NO CONTACT", "NO CONTACT",
                                     "NO CONTACT", "NO CONTACT", "NO CONTACT",
                                     OPEN, CONTACT)
CONTACT_SENSOR_STATE_NAME           = ("MI SWITCH 1", "MI SWITCH 2",
                                     "RAT SWITCH 1", "RAT SWITCH 2",
                                     "MB SWITCH 1", "MB SWITCH 2",
                                     "APXS DOOR SWITCH", "APXS CONTACT SWITCH")

ARTICULATION_DEV_INSTRUMENT_ID     = APXS
END_GROUP                           = START_IDD_ARTICULATION_STATE

/* COORDINATE SYSTEM STATE: INSTRUMENT DEPLOYMENT DEVICE AT THE START*/

GROUP                                = START_IDD_COORDINATE_SYSTEM
COORDINATE_SYSTEM_NAME              = APXS_FRAME
COORDINATE_SYSTEM_INDEX             = (2, 5, 3, 0, 0)
COORDINATE_SYSTEM_INDEX_NAME        = (SITE, DRIVE, IDD, PMA, HGA)
ORIGIN_OFFSET_VECTOR                = (0.0230152, -0.076101, 0.874005)
ORIGIN_ROTATION_QUATERNION          = (0.922297, -0.0165226, -0.0413094,
                                     0.382304)
POSITIVE_AZIMUTH_DIRECTION          = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION        = DOWN
REFERENCE_COORD_SYSTEM_NAME         = ROVER_FRAME
REFERENCE_COORD_SYSTEM_INDEX        = (2, 5, 3, 1, 2)
END_GROUP                           = START_IDD_COORDINATE_SYSTEM

/* ARTICULATION DEVICE STATE: INSTRUMENT DEPLOYMENT DEVICE AT END*/

GROUP                                = STOP_IDD_ARTICULATION_STATE
ARTICULATION_DEVICE_ID              = IDD
ARTICULATION_DEVICE_NAME            = "INSTRUMENT DEPLOYMENT DEVICE"
ARTICULATION_DEVICE_ANGLE           = (0.0230152 <rad>, -0.076101 <rad>,
                                     0.874005 <rad>, 9.4095 <rad>,
                                     0.3467 <rad>, 0.922297 <rad>,
                                     0.0165226 <rad>,
                                     0.0413094 <rad>, 0.38230 <rad>,
                                     0.456 <rad>)
ARTICULATION_DEVICE_ANGLE_NAME      = ("JOINT 1 AZIMUTH-ENCODER",
                                     "JOINT 2 ELEVATION-ENCODER",
                                     "JOINT 3 ELBOW-ENCODER",
                                     "JOINT 4 WRIST-ENCODER",
                                     "JOINT 5 TURRET-ENCODER",
                                     "JOINT 1 AZIMUTH-POTENTIOMETER",
                                     "JOINT 2 ELEVATION-POTENTIOMETER",
                                     "JOINT 3 ELBOW-POTENTIOMETER",
                                     "JOINT 4 WRIST-POTENTIOMETER",
                                     "JOINT 5 TURRET-POTENTIOMETER")

ARTICULATION_DEVICE_MODE           = STOWED
ARTICULATION_DEVICE_TEMP           = (0.922297 <degC>, -0.0165226 <degC>)
ARTICULATION_DEVICE_TEMP_NAME      = ("AZIMUTH JOINT 1", "TURRET JOINT 5")
ARTICULATION_DEV_VECTOR            = (1.23456, 3.4567, 23.456)
ARTICULATION_DEV_VECTOR_NAME       = GRAVITY
CONTACT_SENSOR_STATE               = ("NO CONTACT", "NO CONTACT", "NO CONTACT",

```

```

        "NO CONTACT", "NO CONTACT", "NO CONTACT",
        OPEN, CONTACT)
CONTACT_SENSOR_STATE_NAME = ("MI SWITCH 1", "MI SWITCH 2",
        "RAT SWITCH 1", "RAT SWITCH 2",
        "MB SWITCH 1", "MB SWITCH 2",
        "APXS DOOR SWITCH", "APXS CONTACT SWITCH")
ARTICULATION_DEV_INSTRUMENT_ID = APXS
END_GROUP = STOP_IDD_ARTICULATION_STATE

/* COORDINATE SYSTEM STATE: INSTRUMENT_DEPLOYMENT DEVICE AT END*/

GROUP = STOP_IDD_COORDINATE_SYSTEM
COORDINATE_SYSTEM_NAME = APXS_FRAME
COORDINATE_SYSTEM_INDEX = (2, 5, 3, 0, 0)
COORDINATE_SYSTEM_INDEX_NAME = (SITE, DRIVE, IDD, PMA, HGA)
ORIGIN_OFFSET_VECTOR = (0.0230152, -0.076101, 0.874005)
ORIGIN_ROTATION_QUATERNION = (0.922297, -0.0165226, -0.0413094,
        0.382304)
POSITIVE_AZIMUTH_DIRECTION = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION = DOWN
REFERENCE_COORD_SYSTEM_NAME = ROVER_FRAME
REFERENCE_COORD_SYSTEM_INDEX = (2, 5, 3, 1, 2)
END_GROUP = STOP_IDD_COORDINATE_SYSTEM

OBJECT = MEASUREMENT_TABLE
NAME = APXS_SPECTRA
INTERCHANGE_FORMAT = BINARY
ROWS = 12
COLUMNS = 20
ROW_BYTES = 2560
DESCRIPTION = "Table with 12 APXS measurements."

OBJECT = COLUMN
NAME = XRAY_SAMPLING_DURATION
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 1
BYTES = 2
DESCRIPTION = "Lifetime for x-ray spectrum
        in units of 10 seconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = XRAY_SPECTRUM_ID
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 3
BYTES = 2
DESCRIPTION = "Unique id for the spectrum.
        Value increments by one after
        every cycle start command."
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = XRAY_TC_GAIN
DATA_TYPE = MSB_UNSIGNED_INTEGER
START_BYTE = 5
BYTES = 2
DESCRIPTION = "Temperature correction gain

```

```

        multiplier"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME           = XRAY_TC_LINEAR_TERM
DATA_TYPE      = MSB_UNSIGNED_INTEGER
START_BYTE     = 7
BYTES          = 2
DESCRIPTION    = "Temperature correction linear term"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME           = XRAY_COUNTS
DATA_TYPE      = LSB_UNSIGNED_INTEGER
START_BYTE     = 9
BYTES          = 1014
ITEMS          = 507
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           = XRAY_OVERFLOWS
DATA_TYPE      = LSB_UNSIGNED_INTEGER
START_BYTE     = 1023
BYTES          = 2
DESCRIPTION    = "Number of x-rays with energies
                  greater than full scale."
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME           = ALPHA1_SAMPLING_DURATION
DATA_TYPE      = LSB_UNSIGNED_INTEGER
START_BYTE     = 1025
BYTES          = 2
DESCRIPTION    = "Lifetime for alpha spectrum
                  in units of 10 seconds."
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME           = ALPHA1_SPECTRUM_ID
DATA_TYPE      = LSB_UNSIGNED_INTEGER
START_BYTE     = 1027
BYTES          = 2
DESCRIPTION    = "Unique id for the spectrum.
                  Value increments by one after
                  every cycle start command."
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME           = ALPHA1_TC_GAIN
DATA_TYPE      = MSB_UNSIGNED_INTEGER
START_BYTE     = 1029
BYTES          = 2

```

```

DESCRIPTION                = "Temperature correction gain
                             multiplier"
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA1_TC_LINEAR_TERM
DATA_TYPE                  = MSB_UNSIGNED_INTEGER
START_BYTE                 = 1031
BYTES                      = 2
DESCRIPTION                 = "Temperature correction linear term"
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA1_COUNTS
DATA_TYPE                  = LSB_UNSIGNED_INTEGER
START_BYTE                 = 1033
BYTES                      = 502
ITEMS                      = 251
ITEM_BYTES                 = 2
ITEM_OFFSET                = 2
DESCRIPTION                 = "Number of alpha1 particles
                             detected per channel. Channels
                             represent different energy levels."
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA1_OVERFLOWS
DATA_TYPE                  = LSB_UNSIGNED_INTEGER
START_BYTE                 = 1535
BYTES                      = 2
DESCRIPTION                 = "Number of alpha1 particles with
                             energies greater than full scale."
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA2_SAMPLING_DURATION
DATA_TYPE                  = LSB_UNSIGNED_INTEGER
START_BYTE                 = 1537
BYTES                      = 2
DESCRIPTION                 = "Lifetime for alpha2 spectrum
                             in units of 10 seconds."
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA2_SPECTRUM_ID
DATA_TYPE                  = LSB_UNSIGNED_INTEGER
START_BYTE                 = 1539
BYTES                      = 2
DESCRIPTION                 = "Unique id for the spectrum.
                             Value increments by one after
                             every cycle start command."
END_OBJECT                  = COLUMN

OBJECT                      = COLUMN
NAME                       = ALPHA2_TC_GAIN
DATA_TYPE                  = MSB_UNSIGNED_INTEGER
START_BYTE                 = 1541

```

```

    BYTES = 2
    DESCRIPTION = "Temperature correction gain multiplier"
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = ALPHA2_TC_LINEAR_TERM
DATA_TYPE = MSB_UNSIGNED_INTEGER
START_BYTE = 1543
BYTES = 2
DESCRIPTION = "Temperature correction linear term"
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = ALPHA2_COUNTS
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 1545
BYTES = 502
ITEMS = 251
ITEM_BYTES = 2
ITEM_OFFSET = 2
DESCRIPTION = "Number of alpha2 counts detected per channel. Channels represent different energy levels."
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = ALPHA2_OVERFLOWS
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 2047
BYTES = 2
DESCRIPTION = "Number of alpha2 counts with energies greater than full scale."
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = WEB_TEMPERATURE
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 2049
BYTES = 256
ITEMS = 256
ITEM_BYTES = 1
ITEM_OFFSET = 2
DESCRIPTION = "Scaled WEB temperature."
END_OBJECT = COLUMN

OBJECT = COLUMN
NAME = SENSOR_TEMPERATURE
DATA_TYPE = LSB_UNSIGNED_INTEGER
START_BYTE = 2050
BYTES = 256
ITEMS = 256
ITEM_BYTES = 1
ITEM_OFFSET = 2
DESCRIPTION = "Scaled sensor head temperature."
END_OBJECT = COLUMN
END_OBJECT = MEASUREMENT_TABLE

```

```

OBJECT          = ENGINEERING_TABLE
NAME            = APXS_ENGINEERING_DATA
INTERCHANGE_FORMAT = BINARY
ROWS           = 1
COLUMNS       = 12
ROW_BYTES      = 2048
DESCRIPTION    = "Engineering data from APXS internal
                RAM."

OBJECT          = COLUMN
NAME            = XRAY_TC_GAIN
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 1
BYTES           = 2
DESCRIPTION     = "Temperature correction gain
                multiplier"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = XRAY_TC_LINEAR_TERM
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 3
BYTES           = 2
DESCRIPTION     = "Temperature correction linear term"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ALPHA1_TC_GAIN
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 5
BYTES           = 2
DESCRIPTION     = "Temperature correction gain
                multiplier"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ALPHA1_TC_LINEAR_TERM
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 7
BYTES           = 2
DESCRIPTION     = "Temperature correction linear term"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ALPHA2_TC_GAIN
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 9
BYTES           = 2
DESCRIPTION     = "Temperature correction gain
                multiplier"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ALPHA2_TC_LINEAR_TERM
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 11

```

BYTES	= 2
DESCRIPTION	= "Temperature correction linear term"
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= CYCLE_INTERVAL
DATA_TYPE	= MSB_UNSIGNED_INTEGER
START_BYTE	= 13
BYTES	= 1
DESCRIPTION	= "Cycle interval/minute"
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= SPARE
DATA_TYPE	= UNSIGNED_INTEGER
START_BYTE	= 14
BYTES	= 1
DESCRIPTION	= "Spare byte"
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= RESERVED
DATA_TYPE	= UNSIGNED_INTEGER
START_BYTE	= 15
BYTES	= 9
ITEMS	= 9
ITEM_BYTES	= 1
ITEM_OFFSET	= 1
DESCRIPTION	= "Reserved bytes"
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= UPTIME
DATA_TYPE	= LSB_UNSIGNED_INTEGER
START_BYTE	= 24
BYTES	= 2
DESCRIPTION	= "Uptime from the last power up or software reset/10s."
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= RESERVED
DATA_TYPE	= UNSIGNED_INTEGER
START_BYTE	= 26
BYTES	= 6
ITEMS	= 6
ITEM_BYTES	= 1
ITEM_OFFSET	= 1
DESCRIPTION	= "Reserved bytes"
END_OBJECT	= COLUMN
OBJECT	= COLUMN
NAME	= LOG_BOOK_ADDRESS
DATA_TYPE	= MSB_UNSIGNED_INTEGER
START_BYTE	= 32
BYTES	= 2
DESCRIPTION	= "Address into the LOG_BOOK"

```

                                identifying the location of
                                the last command."
END_OBJECT                      = COLUMN

OBJECT                          = COLUMN
  NAME                          = RESERVED
  DATA_TYPE                     = UNSIGNED_INTEGER
  START_BYTE                     = 34
  BYTES                          = 221
  ITEMS                          = 221
  ITEM_BYTES                     = 1
  ITEM_OFFSET                   = 1
  DESCRIPTION                    = "Reserved bytes"
END_OBJECT                      = COLUMN

OBJECT                          = COLUMN
  NAME                          = LOG_BOOK
  DATA_TYPE                     = UNSIGNED_INTEGER
  START_BYTE                     = 255
  BYTES                          = 1794
  ITEMS                          = 1794
  ITEM_BYTES                     = 1
  ITEM_OFFSET                   = 1
  DESCRIPTION                    = "Circular written log book of
                                received commands.  For every
                                command received by the instrument,
                                the command opcode is written into
                                the log book and the LOG_BOOK_ADDRESS
                                is increased."
END_OBJECT                      = COLUMN
END_OBJECT                      = ENGINEERING_TABLE
END

```

## APPENDIX B – APXS LABEL KEYWORD DEFINITIONS

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>• Location in PDS Label</li> <li>• Source</li> </ul>
APPLICATION_PROCESS_ID	Identifies the source/process which created the data.	Integer		(see APPLICATION_PROCESS_NAME)	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> CCSDS:Primary:APID</p>
APPLICATION_PROCESS_NAME	Provides the name associated with the source/process which created the data.  Note: For Mars Pathfinder, the queues were distinguished on the basis of type and priority of data.	String (256)		APID NAME 21 "PANCAM LEFT" 22 "PANCAM RIGHT" 23 "NAVCAM LEFT" 24 "NAVCAM RIGHT" 25 "HAZCAM LEFT FRONT" 26 "HAZCAM RIGHT FRONT" 27 "HAZCAM LEFT REAR" 28 "HAZCAM RIGHT REAR" 29 "MI" 30 "DESCENT IMAGER" 31 "MINITES" 32 "APXS" 33 "MB" 34 "RAT"	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Table Lookup CCSDS:Primary:APID</p>
APPLICATION_PROCESS_SUBTYPE_ID	Identifies the source/subprocess which created the data.	Integer		0 APXS SCIENCE DATA	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> CCSDS:Primary:subtype</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
ARTICULATION_DEVICE_ANGLE	<p>Provides the value of an angle between two parts or segments of an articulated device.</p> <p>Note: MER uses radians. The PDS default unit for this keyword is degrees, so the &lt;rad&gt; tag is required for MER data.</p>	Float array[10]	radians (<rad> unit tag required)		<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)</p> <p>b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Group Dependent</p> <p>a) IDPH:bgn-&gt;idd.q_enc[5] IDPH:bgn-&gt;idd.q_pot[5]</p> <p>c) IDPH:end-&gt;idd.q_enc[5] IDPH:end-&gt;idd.q_pot[5]</p>
ARTICULATION_DEVICE_ANGLE_NAME	Provides the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE.	String array[10]		("JOINT 1 AZIMUTH-ENCODER", "JOINT 2 ELEVATION-ENCODER", "JOINT 3 ELBOW-ENCODER", "JOINT 4 WRIST-ENCODER", "JOINT 5 TURRET-ENCODER", "JOINT 1 AZIMUTH-POTENTIOMETER", "JOINT 2 ELEVATION-POTENTIOMETER", "JOINT 3 ELBOW-POTENTIOMETER", "JOINT 4 WRIST-POTENTIOMETER", "JOINT 5 TURRET-POTENTIOMETER")	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)</p> <p>b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Static Values</p>
ARTICULATION_DEVICE_ID	Specifies the abbreviated ID of the articulation device described by the containing group. 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.	string		IDD	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)</p> <p>b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Static Values</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
ARTICULATION_DEVICE_MODE	<p>Indicates the deployment state (i.e., physical configuration) of an articulation device at the time of data acquisition.</p> <p>For MER, this is the mode of the last move:</p> <p>“FREE SPACE” – IDD arm was moved where there was no contact with a target expected.</p> <p>“GUARDED” – IDD arm was moved where contact with the target was expected.</p> <p>“RETRACTING” – IDD arm was moved where an instrument is removed from a target.</p> <p>“PRELOAD” – IDD arm stays in contact with the target and applies force or overtravel on an instrument.</p>	String		<u>IDD</u> 0=“FREE SPACE”, 1=“GUARDED”, 2=“RETRACTING”, 3=“PRELOAD”	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)            b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Group Dependent; Table Lookup            a) IDPH:bgn-&gt;idd.idd_mode            b) IDPH:end-&gt;idd.idd_mode</p>
ARTICULATION_DEVICE_NAME	<p>Specifies the common name of the articulation device described by the containing group.</p> <p>Note: The associated ARTICULATION_DEVICE_ID element provides an abbreviated name or acronym for the articulated device.</p>	string		“INSTRUMENT DEPLOYMENT DEVICE”	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)            b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Static Value</p>
ARTICULATION_DEVICE_TEMP	Provides the temperature, in degrees Celsius, of an articulated device or some part of an articulated device.	Float array[2]	deg C (<degC> unit tag required)	-3.4e38 to 3.4e38	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)            b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>a) IDPH:bgn-&gt;idd.temp[2]            b) IDPH:end-&gt;idd.temp[2]</p>
ARTICULATION_DEVICE_TEMP_NAME	An array of the formal names identifying each of the values used in ARTICULATION_DEVICE_TEMP.	string array[2]		(“AZIMUTH JOINT 1”, “TURRET JOINT 5”)	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)            b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>a) Static Value</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
ARTICULATION_DEV_INSTRUMENT_ID	Provides an abbreviated name or acronym which identifies an instrument mounted on the articulation device.	String(12 )		<u>IDD</u> 0 = MI 1 = RAT 2 = MB 3 = APXS	<u>LOCATION</u> a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group)  <u>SOURCE</u> Table Lookup; a) IDPH:bgn->idd.instrument b) IDPH:end->idd.instrument
ARTICULATION_DEV_VECTOR	Provides the direction and magnitude of an external force acting on the articulation device, in the rover's coordinate system at the time the pose was computed.	Float array[3]			<u>LOCATION</u> a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group)  <u>SOURCE</u> a) IDPH:bgn->idd.tilt[3] b) IDPH:end->idd.tilt[3]
ARTICULATION_DEV_VECTOR_NAME	Provides the formal name of the vector type of the articulation device.	String		GRAVITY	<u>LOCATION</u> a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group)  <u>SOURCE</u> Static Value
COMMAND_SEQUENCE_NUMBER	Provides a numeric identifier for a sequence of commands sent to a spacecraft or instrument.  Note: For MER, this is the command number which identifies the specific generating command within the specified sequence.	Integer			<u>LOCATION</u> IDENTIFICATION (Class)  <u>SOURCE</u> UPTH:ProdCmndId; Command Number
CONTACT_SENSOR_STATE	An array of identifiers for the state of an instrument's or instrument host's contact sensors at a specified time.  For MER, "CONTACT" or "NO CONTACT" for most values. For the value corresponding to APXS DOOR SWITCH (entry 7), values are "OPEN" or "CLOSED".	String array[8]		0 = "NO CONTACT" or "CLOSED" 1 = "CONTACT" or "OPEN"  "CONTACT" or "NO CONTACT" for all array positions except for position 7, which would be "OPEN" or "CLOSED"	<u>LOCATION</u> a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group)  <u>SOURCE</u> <u>Lookup Table</u> a) IDPH:bgn->idd.contact (bit map) b) IDPH:end->idd.contact (bit map)

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
CONTACT_SENSOR_STATE_NAME	An array of the formal names identifying each of the values used in CONTACT_SENSOR_STATE.	String array[8]		("MI SWITCH 1", "MI SWITCH 2", "RAT SWITCH 1", "RAT SWITCH 2", "MB SWITCH 1", "MB SWITCH 2", "APXS DOOR SWITCH", "APXS CONTACT SWITCH")	<p><b>LOCATION</b></p> <p>a) START_IDD_ARTICULATION_STATE (Group)</p> <p>b) STOP_IDD_ARTICULATION_STATE (Group)</p> <p><b>SOURCE</b></p> <p>Static Value</p>
COORDINATE_SYSTEM_INDEX	<p>Instance of the coordinate frame in which the values herein are expressed. This is a group of integers that can be used to record and track the movement of a rover during surface operations.</p> <p>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. This index is a set of integers which serve to identify coordinate system instances in a mission-specific manner.</p> <p>For MER, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the rover moves (or may potentially have moved, e.g. due to arm motion). The full counter may have up to 5 values (SITE, DRIVE, IDD, PMA, HGA), but normally only the first value (for SITE frames) or the <b>five</b> values (for LOCAL_LEVEL or ROVER frames) are used for defining coordinate system instances. It is legal to use any number of indices to describe a coordinate system instance, however.</p> <p>Example: COORDINATE_SYSTEM_INDEX = (1,3,2,3,2)</p>	integer array[5]			<p><b>LOCATION</b></p> <p>a) START_IDD_COORDINATE_SYSTEM (Group)</p> <p>b) STOP_IDD_COORDINATE_SYSTEM (Group)</p> <p>c) ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b></p> <p>Calculated from IDPH:bgn-&gt;rmc and COORDINATE_SYSTEM_NAME</p> <p>Use all 5 elements except if the COORDINATE_SYSTEM_NAME = SITE_FRAME, then only use the first element.</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>Location in PDS Label</li> <li>Source</li> </ul>
COORDINATE_SYSTEM_INDEX_NAME	An array of the formal names identifying each integer specified in COORDINATE_SYSTEM_INDEX.	String array[6]		(SITE, DRIVE, IDD, PMA, HGA)	<p><b>LOCATION</b></p> <p>a) START_IDD_COORDINATE_SYSTEM (Group)</p> <p>b) STOP_IDD_COORDINATE_SYSTEM (Group)</p> <p>c) ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b></p> <p>Static Value; Group Dependant</p> <p>Should match the number of values in the COORDINATE_SYSTEM_INDEX</p>
COORDINATE_SYSTEM_NAME	<p>Provides the full name of the coordinate system to which the state vectors are referenced.</p> <p>When in a COORDINATE_SYSTEM_STATE 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 missions), which have multiple instances using the same name, also require COORDINATE_SYSTEM_INDEX to completely identify the coordinate system.</p>	string(30)		<p>ROVER_FRAME, SITE_FRAME, MAST_FRAME, LOCAL_LEVEL_FRAME,</p> <p><u>IDD only</u> 0=MI_FRAME 1=RAT_FRAME 2=MB_FRAME 3=APXS_FRAME,</p>	<p><b>LOCATION</b></p> <p>a) START_IDD_COORDINATE_SYSTEM (Group)</p> <p>b) STOP_IDD_COORDINATE_SYSTEM (Group)</p> <p>c) ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b></p> <p>Group dependant</p> <p>a) one of the IDD only frames based on IDPH:bgn-&gt;idd.instrument</p> <p>b) one of the IDD only frames based on IDPH:end-&gt;idd.instrument</p> <p>c) ROVER_FRAME</p>
DATA_SET_ID	<p>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.</p> <p>Note: In the PDS, the values for both DATA_SET_ID and DATA_SET_NAME are constructed according to standards outlined in the Standards Reference.</p>	string(40)		<p>APXS</p> <p>"MER1-M-APXS-2-EDR-OPS-V1.0",</p> <p>"MER2-M-APXS-2-EDR-OPS-V1.0"</p> <p>"SIM1-M-APXS-2-EDR-V1.0"</p> <p>"SIM2-M-APXS-2-EDR-V1.0"</p>	<p><b>LOCATION</b></p> <p>IDENTIFICATION (Class)</p> <p><b>SOURCE</b></p> <p>PDS</p> <p>Table Lookup</p> <p>CCSDS:Primary:APID</p> <p>CHDO-82:scft_id</p>
EARTH_RECEIVED_START_TIME	Provides the beginning time at which telemetry was received during a time period of interest. This should be represented in UTC system format.	Datetime		YYYY-MM-DDThh:mm:ss[.fff]	<p><b>LOCATION</b></p> <p>TELEMETRY (Class)</p> <p><b>SOURCE</b></p> <p>Calculated</p> <p>CHDO_82:ert</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
EARTH_RECEIVED_STOP_TIME	Provides the ending time for receiving telemetry during a time period of interest. This should be represented in UTC system format.	Datetime		YYYY-MM-DDThh:mm:ss[.fff]	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Calculated CHDO_82:ert</p>
EXPECTED_PACKETS	Provides the total number of telemetry packets which constitute a complete data product, i.e., a data product without missing data.  For MER, "Packets" are also referred to as "Parts".	Integer			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> UPTH:TotalParts</p>
FILE_RECORDS	Indicates 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.	Integer		5	<p><b>LOCATION</b> FILE DATA ELEMENT (Class)</p> <p><b>SOURCE</b> Static Value</p>
INSTRUMENT_HOST_ID	Provides a unique identifier for the host where an instrument is 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.	String(6)		<p><b>SCID</b>      <b>Keyword Values</b></p> <p>253          "MER1"</p> <p>254          "MER2"</p> <p>252          "SIM1"</p> <p>255          "SIM2"</p>	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Table Lookup CHDO_82:scft_id</p>
INSTRUMENT_HOST_NAME	Provides 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.	String		<p>"MARS EXPLORATION ROVER 1"</p> <p>"MARS EXPLORATION ROVER 2"</p> <p>"SIMULATED MARS EXPLORATION ROVER 1"</p> <p>"SIMULATED MARS EXPLORATION ROVER 2"</p>	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Table Lookup CHDO_82:scft_id</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>• Location in PDS Label</li> <li>• Source</li> </ul>
INSTRUMENT_ID	<p>Provides an abbreviated name or acronym which identifies an instrument.</p> <p>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.</p> <p>Example values: IRTM (for Viking Infrared Thermal Mapper), PWS (for plasma wave spectrometer).</p>	string(12)		<p><u>IDD</u></p> <p>29 = "MI"</p> <p>34 = "RAT"</p> <p>33 = "MB"</p> <p>32 = "APXS"</p>	<p><b><u>LOCATION</u></b> IDENTIFICATION (Class)</p> <p><b><u>SOURCE</u></b> Table Lookup; CCSDS:Primary:APID</p>
INSTRUMENT_TYPE	<p>Identifies the type of an instrument.</p> <p>Example values: POLARIMETER, RADIOMETER, REFLECTANCE SPECTROMETER, VIDICON CAMERA.</p> <p>Note that mosaics may contain more than one value in an array.</p>	String		"SPECTROMETER"	<p><b><u>LOCATION</u></b> IDENTIFICATION (Class)</p> <p><b><u>SOURCE</u></b> Static Value</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
LOCAL_TRUE_SOLAR_TIME	<p>Local true solar time, or LTST, 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.</p> <p>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. )</p> <p>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,</p> $LTST = ( a(P) - a(TS) ) * ( 24 / 360 ) + 12$ <p>where, LTST = the local true solar time in true solar hours</p> <p>a( P ) = ARA of the point on the planet' s surface in deg</p> <p>a( TS ) = ARA of the true sun in deg</p>	string(12)		NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	<p><u>LOCATION</u></p> <ul style="list-style-type: none"> <li>•IDENTIFICATION ( Class)</li> </ul> <p><u>SOURCE</u></p> <ul style="list-style-type: none"> <li>•Calculation:</li> <li>- IDPH:bgn-&gt;rover_time.seconds</li> <li>- IDPH:bgn-&gt;rover_time.subseconds</li> <li>- SCLK Kernel</li> <li>- Landing- Site kernel</li> <li>- P Kernel</li> </ul>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>Location in PDS Label</li> <li>Source</li> </ul>
MAGNET_ID	Identifies a magnet instrument that is visible in an image or observation.	String		"N/A", "CAPTURE", "FILTER", "UNK"	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Calculated via an algorithm using ??????????</p>
MISSION_NAME	Identifies 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.	String		"MARS EXPLORATION ROVER"	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Static Value</p>
MISSION_PHASE_NAME	Provides the commonly-used identifier of a mission phase.	string(30)		"CRUISE", "EXTENDED MISSION", "PRIMARY MISSION", "ATLO", "ORT1", "ORT2", "ORT3", "ORT4", "TBD"	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Operator Supplied Parameter</p>
OBSERVATION_ID	Uniquely identifies 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.	String			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> IPDH:bgn-&gt;context_id</p>
ORIGIN_OFFSET_VECTOR	Specifies the offset from the reference coordinate system's origin to the origin of the coordinate system being defined by the enclosing COORDINATE_SYSTEM_STATE group. In other words, it is the location of the current system's origin as measured in the reference system.  For MER, here is an example: In the case of the PMA_COORDINATE_SYSTEM group, ORIGIN_OFFSET_VECTOR describes the rotation of the PMA (camera head) boresight (about the ORIGIN_OFFSET_VECTOR) relative to the Rover frame.	Float array[3]	meters		<p><b>LOCATION</b> a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b> a) IDPH:bgn-&gt;idd.pos[3] b) IDPH:end-&gt;idd.pos[3] c) IDPH:bgn-&gt;pos.v[3]</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
ORIGIN_ROTATION_QUATERNION	<p>Provides an array of four values that specifies the rotation of the coordinate system being defined by the enclosing COORDINATE_SYSTEM_STATE 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.</p> <p>Quaternions are expressed as a set of four numbers in the order                      (s, v1, v2, v3),                      where  <math>s = \cos(\theta/2)</math>  <math>v(n) = \sin(\theta/2) \cdot a(n)</math>.                      theta = the angle of rotation                      a = the (x,y,z) vector around which the rotation occurs.</p> <p>For MER, 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 declared to be "UP" or "DOWN"</p> <p>For MER, here is an example:                      In the case of the PMA_COORDINATE_SYSTEM group, ORIGIN_OFFSET_VECTOR describes the rotation of the PMA (camera head) boresight (about the ORIGIN_OFFSET_VECTOR) relative to the Rover frame.</p>	Float array[4]			<p><b>LOCATION</b>                      a) START_IDD_COORDINATE_SYSTEM (Group)                      b) STOP_IDD_COORDINATE_SYSTEM (Group)                      c) ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b>                      a) IDPH:bgn-&gt;idd.quaternion[4]                      b) IDPH:end-&gt;idd.quaternion[4]                      c) IDPH:bgn-&gt;pos.q[4]</p> <p>Received in (v1, v2, v3, s) order.</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
PACKET_MAP_MASK	<p>A binary or hexadecimal number identifying which of a data file's expected packets were actually received. The digits correspond positionally with the relative packet numbers of the data file. The bits are to be read left to right; i.e., the first (left-most) digit of the number corresponds to the first packet of the data file. A bit value of 1 indicates that the packet was received; a value of 0 indicates that it was not received.</p> <p>The number is stored in the PDS radix notation of &lt;radix&gt;#&lt;value&gt;#.</p>	Non-decimal			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Calculated UPDTH:PartNumber</p>
PDS_VERSION_ID	<p>Represents 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'.</p> <p>Examples: PDS3, PDS4.</p>	String(6)		"PDS3"	<p><b>LOCATION</b> PDS required</p> <p><b>SOURCE</b> PDS</p>
PLANET_DAY_NUMBER	<p>Indicates 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 (which is day zero).</p>	Integer		NOTE: Value will be uncalibrated if SPICE kernels are not available.	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Calculated IPDH:bgn-&gt;rover_time.seconds IPDH:bgn-&gt;rover_time.subseconds SCLK kernel</p>
POSITIVE_AZIMUTH_DIRECTION	<p>Provides 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 CLOCKWISE indicates that Azimuth is measured positively Clockwise, and COUNTERCLOCKWISE indicates that Azimuth increases positively Counter-clockwise.</p>	String		"CLOCKWISE", "COUNTERCLOCKWISE"	<p><b>LOCATION</b> a) ROVER_COORDINATE_SYSTEM (Group) b) START_IDD_COORDINATE_SYSTEM (Group) c) STOP_IDD_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b> Static Value</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
POSITIVE_ELEVATION_DIRECTION	<p>Provides 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.</p> <p>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.</p> <p>For MER, which follows the Mars Pathfinder convention, increasing elevation ("UP") moves towards the negative Z axis.</p>	String		"UP", "DOWN"	<p><b>LOCATION</b></p> <p>a) ROVER_COORDINATE_SYSTEM (Group)</p> <p>b) START_IDD_COORDINATE_SYSTEM (Group)</p> <p>c) STOP_IDD_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b></p> <p>Static Value</p>
PROCESSING_HISTORY_TEXT	Provides an entry for each processing step and program used in generating a particular data file.	string		"CODMAC LEVEL 1 TO LEVEL 2 CONVERSION VIA JPL/MIPL MERTELEMPROC"	<p><b>LOCATION</b></p> <p>HISTORY (Class)</p> <p><b>SOURCE</b></p> <p>Static Value</p>
PRODUCER_INSTITUTION_NAME	Identifies 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.	string(60)		"MULTIMISSION IMAGE PROCESSING SUBSYSTEM, JET PROPULSION LAB"	<p><b>LOCATION</b></p> <p>IDENTIFICATION (Class)</p> <p><b>SOURCE</b></p> <p>Static Value</p>
PRODUCT_CREATION_TIME	Defines the UTC system format time when a product was created. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]Z	string			<p><b>LOCATION</b></p> <p>IDENTIFICATION (Class)</p> <p><b>SOURCE</b></p> <p>Calculated</p>
PRODUCT_ID	<p>Represents a permanent, unique identifier assigned to a data product by its producer. See also: source_product_id.</p> <p>Note: In the PDS, the value assigned to product_id must be unique within its data set.</p> <p>Additional note: The product_id can describe the lowest-level data object that has a PDS label.</p>	string(40)			<p><b>LOCATION</b></p> <p>IDENTIFICATION (Class)</p> <p><b>SOURCE</b></p> <p>Calculated</p> <p>Filename less the extension</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>• Location in PDS Label</li> <li>• Source</li> </ul>
PRODUCT_TYPE	Identifies the type or category of a data product within a data set.	String(8)		"MB_EDR" "APXS_EDR"	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Static Value</p>
PRODUCT_VERSION_ID	<p>Identifies the version of an individual product within a data set.</p> <p>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.</p> <p>For MER, PRODUCT_VERSION_ID includes a Version field that begins with "V" followed by the Version decimal number of the controlling SIS document.</p> <p>Example: "V2.0 D-22848"</p> <p>Note: This might not be the same as the data set version that is an element of the DATA_SET_ID value.</p>	string(12)		"V<vernum> D-22848"	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> User Parameter</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>Location in PDS Label</li> <li>Source</li> </ul>
QUATERNION_MEASUREMENT_METHOD	<p>Specifies the quality of the rover orientation estimate.</p> <p>“UNKNOWN” – The attitude should simply not be trusted. This is the initial grade given on Landing, for example.</p> <p>“TILT_ONLY” – The attitude estimate is only good for tilt determination (2-axis knowledge). Activities which require azimuth knowledge should be careful.</p> <p>“COARSE” – The attitude estimate is “complete” (it has all three axes) but is crude. This can occur because a sungaze has not yet been performed or because some event (such as traverses or IDD activity) have reduced the quality of the estimate (a.k.a. “ThreeAxisCoarse”).</p> <p>“FINE” – Sungaze completed successfully, and the attitude estimate is sufficient for pointing HGA (a.k.a. “ThreeAxisFine”).</p>	String		0 = “UNKNOWN” 1 = “TILT_ONLY” 2 = “COARSE” 3 = “FINE”	<p><b>LOCATION</b> ROVER_COORDINATE_SYSTEM (Group)</p> <p><b>SOURCE</b> Table Lookup a) IDPH:bgn-&gt;att_qual</p>
RECEIVED_PACKETS	Provides the total number of telemetry packets which constitute a reconstructed data product.	Integer			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Calculated UPTH:PacketNumber</p>
RECORD_BYTES	<p>Indicates the number of bytes in a physical file record, including record terminators and separators.</p> <p>Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.</p>	Integer		0 to n	<p><b>LOCATION</b> FILE (Class)</p> <p><b>SOURCE</b> Calculated</p>
RECORD_TYPE	<p>Indicates the record format of a file.</p> <p>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.</p>	string(20)		“FIXED_LENGTH”	<p><b>LOCATION</b> FILE (Class)</p> <p><b>SOURCE</b> Static Value</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
REFERENCE_COORD_SYSTEM_INDEX	<p>Identifies 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.</p> <p>For MER, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the rover moves (or may potentially have moved, e.g. due to arm motion). The full counter may have up to 5 values (SITE, DRIVE, IDD, PMA, HGA), but normally only the first value (for SITE frames) or the five values (for LOCAL_LEVEL or ROVER frames) are used for defining reference coordinate system instances. It is legal to use any number of indices to describe a reference coordinate system instance, however.</p> <p>See also REFERENCE_COORD_SYSTEM_NAME and COORDINATE_SYSTEM_INDEX.</p>	Integer array[5]			<p><b>LOCATION</b></p> <ul style="list-style-type: none"> <li>a) START_IDD_COORDINATE_SYSTEM (Group)</li> <li>b) STOP_IDD_COORDINATE_SYSTEM (Group)</li> <li>c) ROVER_COORDINATE_SYSTEM (Group)</li> </ul> <p><b>SOURCE</b></p> <p># of values used is group dependent</p> <ul style="list-style-type: none"> <li>a) IDPH:bgn-&gt;rmc(0,1)</li> <li>b) IDPH:bgn-&gt;rmc(0,1)</li> <li>c) IDPH:bgn-&gt;rmc(0)</li> </ul> <p>Calculated from IDPH:bgn-&gt;rmc and REFERENCE_COORD_SYSTEM_NAME</p> <p>Use all five elements except if the REFERENCE_COORD_SYSTEM_NAME = SITE_FRAME, then only use the first element.</p>
REFERENCE_COORD_SYSTEM_NAME	<p>Provides the full name of the reference coordinate system 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. Non-unique coordinate systems (such as "SITE" for rover missions), which have multiple instances using the same name, also require REFERENCE_COORD_SYSTEM_INDEX to completely identify the reference coordinate system.</p> <p>For MER, the reference is usually a SITE frame.</p>	string(20)		ROVER_FRAME, SITE_FRAME, MAST_FRAME, PANCAM_FRAME, LOCAL_LEVEL_FRAME, LANDER_FRAME	<p><b>LOCATION</b></p> <ul style="list-style-type: none"> <li>a) START_IDD_COORDINATE_SYSTEM (Group)</li> <li>b) STOP_IDD_COORDINATE_SYSTEM (Group)</li> <li>c) ROVER_COORDINATE_SYSTEM (Group)</li> </ul> <p><b>SOURCE</b></p> <p># of values used is group dependent</p> <ul style="list-style-type: none"> <li>a) "ROVER_FRAME"</li> <li>b) "ROVER_FRAME"</li> <li>c) "SITE_FRAME"</li> </ul>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
RELEASE_ID	<p>Unique identifier associated with the release to the public of all or part of a data set. The first release of a data set should have a RELEASE_ID of "0001"</p> <p>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. For each Rover mission, the first release of a data set should have a value of "0001"</p> <p>For example, the first release of the Pancam EDR data set on MER-1 will be August 3, 2004 (according to the Archive Plan), so those products will have RELEASE_ID = "0001". The next Pancam EDR release will be October 4, 2004, so those products will have RELEASE_ID = "0002". The Pancam EDRs from the other rover are a separate data set. Those will be released August 24, 2004 (RELEASE_ID = "0001") and October 25, 2004 (RELEASE_ID = "0002").</p>	String			<p><b>LOCATION</b> IDENTIFICATION</p> <p><b>SOURCE</b> User Parameter</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
ROVER_MOTION_COUNTER	<p>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. MER), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values.</p> <p>For MER, the motion counter consists of five values. In order, they are Site, Drive, IDD, PMA, and HGA. 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 IDD, PMA, and HGA increment whenever the corresponding articulation device moves. It is TBD whether IDD, PMA, and HGA are independent of each other, or reset the others to 0 in a hierarchical manner when they are incremented.</p> <p>Conceptually, a sixth value could be added by ground processing to indicate unintentional slippage (e.g. the wind blew the rover off a rock). This sixth value will never occur in telemetry but might occur in certain RDR's. (implementation of this is TBD).</p>	Integer array[5]			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> IDPH: bgn-&gt;rmc(5)</p>
ROVER_MOTION_COUNTER_NAME	An array that provides the formal names identifying each integer in ROVER_MOTION_COUNTER.	String array[5]		(SITE, DRIVE, IDD, PMA, HGA)	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Static Value</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
SAMPLING_COUNT	The sampling_count element provides the number of data samples taken by an instrument or detector.	integer			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> IDPH:telemetry_count</p>
SEQUENCE_ID	Provides an identification of the spacecraft sequence associated with the given product. This element replaces the older seq_id, which should no longer be used.	string(30)			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> UPTH:PrdCmdID:SequenceID</p>
SEQUENCE_VERSION_ID	Provides the version identifier for a particular observation sequence used during planning or data processing.	string(30)			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> UPTH:PrdCmdID:SequenceVersion</p>
SOFTWARE_NAME	Identifies data processing software such as a program or a program library.	string(60)		"MERTELEMPROC"	<p><b>LOCATION</b> HISTORY (Class)</p> <p><b>SOURCE</b> User Parameter</p>
SOFTWARE_VERSION_ID	Indicates the version (development level) of a program or a program library.	string(20)			<p><b>LOCATION</b> HISTORY (Class)</p> <p><b>SOURCE</b> User Parameter</p>
SPACECRAFT_CLOCK_CNT_PARTITION	Indicates the clock partition active for the SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT elements.	integer		1	<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Static Value</p>
SPACECRAFT_CLOCK_START_COUNT	<p>Provides the value of the spacecraft clock at the beginning of a time period of interest.</p> <p>Format is dddddddd.ddd, measured in units of seconds and stored internally as a floating point number.</p> <p>Note: In the PDS, sclk_start_counts have been represented in the following ways: Voyager - Flight Data Subsystem (FDS) clock count (floating point 7.2) Mariner 9 - Data Automation Subsystem, Mariner 10 - FDS - spacecraft_clock</p>	string(30)			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> IDPH:bgn-&gt;rover_time.seconds IDPH:bgn-&gt;rover_time.subseconds</p> <p>Note: It is possible for the sclk to be earlier than the DVT time reported in the data product meta data file and the sclk reported in the data product file name.</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
SPACECRAFT_CLOCK_STOP_COUNT	Provides the value of the spacecraft clock at the end of a time period of interest.  Format is dddddd.ddd, measured in units of seconds and stored internally as a floating point number.	string(30)			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> IDPH:end-&gt;rover_time.seconds IDPH:end-&gt;rover-time.subseconds</p>
SPICE_FILE_NAME	Provides the names of the SPICE files used in processing the data. For Galileo, the SPICE files are used to determine navigation and lighting information.	string (180)			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> User parameter</p>
START_TIME	Provides 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. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]	string			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Calculated IDPH:bgn-&gt;rover_time.seconds IDPH:bgn-&gt;rover_time.subseconds SCLK kernel</p>
STOP_TIME	Provides 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. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]	string			<p><b>LOCATION</b> IDENTIFICATION (Class)</p> <p><b>SOURCE</b> Calculated IDPH:end-&gt;rover_time.seconds IDPH:end-&gt;rover_time.subseconds IDPH:ImgTImHdr:exp_time SCLK kernel</p>
TARGET_NAME	Identifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE.	string(30)		"MARS", "CALIBRATION"	<p><b>LOCATION</b> a) IDENTIFICATION (Class)</p> <p><b>SOURCE</b> a) MARS</p>
TARGET_TYPE	Identifies the type of a named target.	string		"CALIBRATION", "DUST", "N/A", "SUN", "PLANET"	<p><b>LOCATION</b> a) IDENTIFICATION (Class)</p> <p><b>SOURCE</b> a) Static Value – PLANET</p>

Keyword Name	Definition	Type	Units	Valid Values	<ul style="list-style-type: none"> <li>● Location in PDS Label</li> <li>● Source</li> </ul>
TELEMETRY_FORMAT_ID	A telemetry format code	String		APXS 100="SPT" – APXS Spectra	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Table Lookup IDPH: flavor</p>
TELEMETRY_PROVIDER_ID	Identifies the provider and version of the telemetry data used in the generation of this data.	string(3)		"SSW MER_DP" "TTACS"	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> User Parameter</p>
TELEMETRY_SOURCE_NAME	Identifies the name of the telemetry source used in the creation of this data set.	string			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> Name of the input data product.</p>
TELEMETRY_SOURCE_TYPE	Classifies the source of the telemetry used in creation of this data set.	string(12)		"DATA PRODUCT", "SFDU"	<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> User Parameter</p>
TLM_INST_DATA_HEADER_ID	Indicates the version of the instrument specific information provided with telemetry data products. Incremented by FSW whenever there is a change to the header structure.	Integer			<p><b>LOCATION</b> TELEMETRY (Class)</p> <p><b>SOURCE</b> IDPH: version</p>