Mars Exploration Rover (MER)

Software Interface Specification

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

Mission: MER Date: September 27, 2005

Module ID: SIS-SCI009-MER

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

Signatures

GDS Generating Elements:

Ops Product Generation Subsystem (OPGS)
Justin Maki

Subsystem Engineer Date

Concurrance:

Mössbauer Payload Element Lead
Göstar Klingelhöfer,

PEL Date

MER Mössbauer Investigation Scientist
Albert Yen

Investigation Scientist Date

MER Science Manager
John Callas

Manager Date

PDS Geosciences Node Manager
Raymond Arvidson

Manager Date

MER MIPL Cognizant Engineer
Doug Alexander

Cognizant Engineer Date

Mars Exploration Rover (MER) Project

MB EDR Software Interface Specification (SIS)

Version 2.1

Prepared by:
Edward A. Guinness

MER MB SIS Custodian
Helen Mortensen

JPL D-22849

SEPTEMBER 27, 2005

Jet Propulsion Laboratory
California Institute of Technology

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
2/8/02	All	First draft	Draft v0.1
6/1/02	All	Second draft	Draft v0.2
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
11/18/02	Cover Page, Appendix B	Replaced Craig Leff with Arthur Amador, fixed rover motion counter number of values, fixed misspelled word COURSE.	Version 1.1.
11/25/02	2.3.2, 2.3.3, 2.3.4, Appendix A	Update to reference data products instead of telemetry packets and the meta-data database. Added Mainz, Germany sites to "who" in the filenaming convention. Updated COORDINATE_SYSTEM_INDEX and INDEX_NAME values in the example.	Version 1.2
3/7/03	2.3.4, Appendix A & B	Removed SAP as receiving element. Updated filenaming convention, Changed BEGIN to START and END to STOP, added RELEASE_ID, added SPICE keywords	Version 2.0
3/28/03	Appendix A& B	Add Z to all times, updated value to PRODUCT_VERSION_ID	Version 2.0
6/13/03	2.3.4, Appendix A & B	Filenaming convention updated. Updated label and definitions. Removed SPICE_FILE_ID	Version 2.0
10/7/03	3.2, Appendix A & B	Added SAMPLING_COUNT to address the TBD item of needing new keyword for TELEMETRY_COUNT. Added LOCAL_TRUE_SOLAR_TIME. Update the Board Sensor temperature conversion. Deleted TLM_CMD_DISCREPANCY_FLAG keyword.	Version 2.0
11/7/03	Figure 1, 2.3.4, 3.3.1, new Appendix B	Update the coordinate system diagram. Added new filenaming convention and label for return of single blocks.	Version 2.0
7/31/04	Appendix A	For archive data, added double quotes to RELEASE_ID,	VERSION 2.01

		LOCAL_TRUE_SOLAR_TIME, OBSERVATION_ID, SEQUENCE_VERSION_ID, SPACECRAFT_CLOCK_START_COUNT, SPACECRAFT_CLOCK_STOP_COUNT. Removed double quotes from PRODUCT_CREATION_TIME, START_TIME, STOP_TIME, EARTH_RECEIVED_START_TIME, EARTH_RECEIVED_STOP_TIME. Added OPS to DATA_SET_ID.	
9/4/05	Signature cover page	Removed Receiving GDS Element (Deborah Bass) and GDS (Frank Singleton) signatures, because they are no longer relevant.	Rev. 2.1
9/4/05	3.2	Removed this sentence: The structure of the logbook is described in the Mössbauer Spectrometer Information Interface Control Document [9]. Removed reference [9] from the reference list. Added these sentences: The logbook contains information on the engineering status of the instrument during a measurement. The information in the logbook is not needed and not useful for calibration or science data analysis.	Rev. 2.1
9/4/05	3.2	Typo in formula for board sensor temperature corrected. The last plus sign was changed to a minus sign. Added location of electronics board (located inside the rover's Warm Electronics Box)	Rev 2.1
9/4/05	3.2	Added: The instrument has four counters to store eight measurements from the detectors (4 detectors with 2 energies each). As a result, the measurements from pairs of detectors are added together and stored in one counter. These four summed spectra are included in the EDR data products. In the generation of the Reduced Data Record (RDR) products, the two 6.4 keV spectra are summed, as are the two 14.4 keV. These summed spectra exist, and are labeled as separate products in the RDR data set.	Rev 2.1
9/4/05	2.2	Added: The compressed spectra are contingency spectra to protect against certain instrument anomalies. They are	Rev 2.1

		not intended for science analysis and are not needed for any science analysis.	
9/4/05	3.2	Added: The MB spectrum stored at location 5400h-71FFh in the fifth block is for backup (a copy of data from one of main temperature windows) and contains no new data.	Rev 2.1
9/4/05	3.2	Added: The integration times for each MB measurement are listed in the observation table provided to the PDS. This table can be found in the document directory of both the EDR and RDR data sets. Integration time is calculated by division of drive cycles by drive frequency. As noted, the number of drive cycles is included with each spectrum in the EDR data files. The drive frequency is about 25 Hz. It can also be derived from the FG_PRESCALER parameter stored in the instrument parameter block. The stored FG_PRESCALER value is converted to drive frequency as follows: fdrive = 900 Hz / FG_PRESCALER. So, for example, a prescaler value of 37 yields a drive frequency of 900/37 = 24.3 Hz.	Rev 2.1

TBD ITEMS

SECTION	DESCRIPTION

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
APSS	Activity Planning and Sequencing Subsystem
CODMAC	Committee on Data Management and Computation
EDR	Experiment Data Record
EEPROM	Electronically Erasable Programmable Read-Only Memory
FEI	File Exchange Interface
FRAM	Ferroelectric Random Access Memory
ICD	Interface Control Document
IDD	Instrument Deployment Device
ISO	International Standards Organization
JPL	Jet Propulsion Laboratory
Kbytes	Kilobytes
LSB	Least Significant Byte
MB	Mössbauer Spectrometer
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	Operation 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
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SRAM	Static Random Access Memory
SSW	System Software
TBD	To Be Determined
TDS	Telemetry Delivery Subsystem

URL	Universal Resource Locator

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 Mössbauer Spectrometer (MB) Experiment Data Record (EDR) with a detailed description of the product and a description of how it was generated, including data sources and destinations. A MB EDR contains data from a measurement session. The MB EDR data are stored in binary format. The MB science team will produce a set of MB Reduced Data Record (RDR) products in ASCII format. The RDR's will be described in a separate SIS document.

This SIS is intended to provide enough information to enable users to understand the MB 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 MB 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, along with the definitions of the keywords in the label.

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, Rev. D, August 28, 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 MB 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
MB EDRs	P	OPGS/MIPL
Mertelemproc	S	MIPL
MIPL database schema	P	MIPL
MB RDRs	P	MB Science Team
MB RDR SIS	D	MB Science Team
Other MB Programs/Products/Documents	P/S/D	MB Science Team
mb2asc	S	Geosciences Node

2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

2.1 Instrument Overview

The MER Mössbauer (MB) spectrometer is designed to characterize the oxidation states of iron and the relative abundance of iron-bearing mineral phases on the martian surface. The MB spectrometer uses a vibrationally-modulated ^{57}Co source to illuminate a target with gamma rays. Backscattered gamma radiation is binned by source velocity to determine the hyperfine splitting of ^{57}Fe nuclear levels. Targets for Mössbauer measurements include soils and rocks (including rocks that have been abraded with the Rock Abrasion Tool), rover-mounted magnets designed to capture atmospheric dust, and a magnetite-rich calibration target mounted on the rover. 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 the ^{57}Co radiation source, the Mössbauer drive unit, a radiation collimator, x-ray and gamma ray detectors, temperature sensors, a calibration source, and contact plate and switches. The MB has three temperature sensors that monitor the temperature of the electronics board, sensor head, and target sample.

Mössbauer measurements are made by placing the instrument directly on a target. Each measurement will usually take approximately 12 hours. Shorter integration times can be used to analyze the major iron minerals in a target. Mössbauer spectra are temperature dependent. Therefore, spectra are collected in thirteen temperature windows. The temperature limits for the windows can be changed with commands sent to the instrument.

2.2 Data Product Overview

Each MB 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 MB EDR binary data file is a copy of the Mössbauer memory buffer. That is, the EDR consists of unprocessed experiment data stored in binary format. The instrument has three sections of memory: SRAM, FRAM, and EEPROM. These memory sections comprise 5 blocks of data, each of which is 32 kilobytes in size. The SRAM, which is volatile memory, contains temperature dependent Mössbauer spectra, drive error signal data an energy spectrum, and temperature data. Mössbauer spectra are measured as counts per channel, where the channels represent velocity bins. The drive error signal measures the difference between the actual and expected velocities of the MB drive unit. The energy spectrum is collected during each MB observation to support a ground analysis of detector performance. The FRAM contains copies of the instrument parameter block and logbook. The EEPROM contains compressed Mössbauer spectra as a backup in case SRAM data are lost. The compressed spectra are contingency spectra to protect against certain instrument anomalies. They are not intended for science analysis and are not needed for any science analysis.

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 EDR data products. MB 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.
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).

NASA	CODMAC	Description
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 MB 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 generated by the SSW team 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 MB EDR. If telemetry data is missing 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.

2.3.3 Data Flow

The MB EDR data products generated by MIPL during operations are created collectively from: a) SSW data products and b) SPICE kernels. They are created on the OSS and then deposited into MIPL's File Exchange Interface (FEI) for electronic distribution to remote sites via a secure subscription protocol. After a data validation period, the MB 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 Mössbauer Spectrometer EDR binary data file is 0.160 MB. The Mössbauer Spectrometer EDR will be generated 60 seconds after the last packet SFDU for the EDR has been received by MIPL from the MER TDS. Mössbauer Spectrometer data will be reprocessed only if packets in the original downlink are not received. Partial files are created with missing data filled with zeroes. The MB EDR will be reprocessed after all data is retransmitted and received and the original version will be overwritten and placed into FEI for distribution.

2.3.4 Labeling and Identification

There is a file naming scheme adapted for the MER image and non-image data products. The scheme applies to the EDR and several RDR data products. The file naming scheme adheres to the Level II 27.3 filename convention to be compliant with PDS standards.

Each MER EDR or RDR data product can be uniquely identified by incorporating into the product filename the Rover Mission identifier, the Instrument identifier, the Starting Spacecraft Clock count (SCLK) of the camera event, the data Product Type, the Site location, the rover

Position within the site, the Sequence number, the camera "Eye", the spectral Filter, the product Creator identifier and a Version number. For non-camera data, several fields do not apply.

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

<rover><inst><sclk><prod><site><pos><seq><eye><filt><who><ver>.<ext>

where,

rover = (1 integer) MER rover mission identifier. Valid values are "**1**" (MER-1) or "**2**" (MER-2), **"3"** (SIM-1), or **"4"** (SIM-2)

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

Valid values for MER instruments:

"**A**" - APXS
"**B**" - Mössbauer
"**T**" - Mini-TES
"**D**" - RAT ("D" for Drill)

Valid values for MER camera instruments not described in this SIS:

"**P**" - Pancam
"**N**" - Navcam
"**F**" - Front Hazcam
"**R**" - Rear Hazcam
"**M**" - Microscopic Imager
"**E**" - EDLcam

sclk = (9 integers) Spacecraft Clock time of the DVT (Data Validity Time)

prod = (3 alpha characters) Product type. Indicates the product to be an EDR or one of several types of Non-projected RDRs. All product types that begin with "**E**" denote a type of EDR, while all other product types denote a type of Non-projected RDR. All product types that end with "**C**" denote CAHV-linearization.

Valid values for MER non-camera instrument products:

Data Product	Value
MB EDR (contains all 5 block)	"**EDR**"
MB Block 1 only EDR	"**ED1**"
MB Block 2 only EDR	"**ED2**"
MB Block 3 only EDR	**"ED3"**
MB Block 4 only EDR	**"ED4"**
MB Block 5 only EDR	**"ED5"**
MB Block unkown EDR	**"EDX"**

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

The valid values,in their progression,are as follows:
Range 0 thru 99 - **"00 ","01 ","02 "… "99 "**
Range 100 thru 1035 - **"A0 ","A1 " … "A9 ","AA ","AB "…"AZ ","B0 ","B1 "… "ZZ "**
Range 1036 thru 1295 - **"0A ","0B "…"0Z ","1A ","1B "…"9Z "**
Range 1296 or greater - **"-##"** (2 pound signs)

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

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

The valid values,in their progression,are as follows:
Range 0 thru 99 - **"00 ","01 ","02 "… "99 "**
Range 100 thru 1035 - **"A0 ","A1 " … "A9 ","AA ","AB "…"AZ ","B0 ","B1 "… "ZZ "**
Range 1036 thru 1295 - **"0A ","0B "…"0Z ","1A ","1B "…"9Z "**
Range 1296 or greater - **"##"** (2 pound signs)

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

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 Gutenburg 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) 1B123456789EDR0103N0062N0M1.DAT Rover MER-1, Mössbauer instrument, EDR, Site 01, Position 03, Seq N0062, produced by MIPL, product version 1.

2.4 Standards Used in Generating Products

2.4.1 PDS Standards

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

2.4.2 Time Standards

The PDS label of a MB 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 description. See Appendix C.

2.4.3 Coordinate System Standards

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 [Ref 5].

Table 3: Coordinate Frames Used for MER Surface Operations

IMAGING-RELATED COORDINATE SYSTEM		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

IMAGING-RELATED COORDINATE SYSTEM		COORDINATE SYSTEM ORIGIN	COORDINATE SYSTEM ORIENTATION
NAME	LABEL KEYWORD		
			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

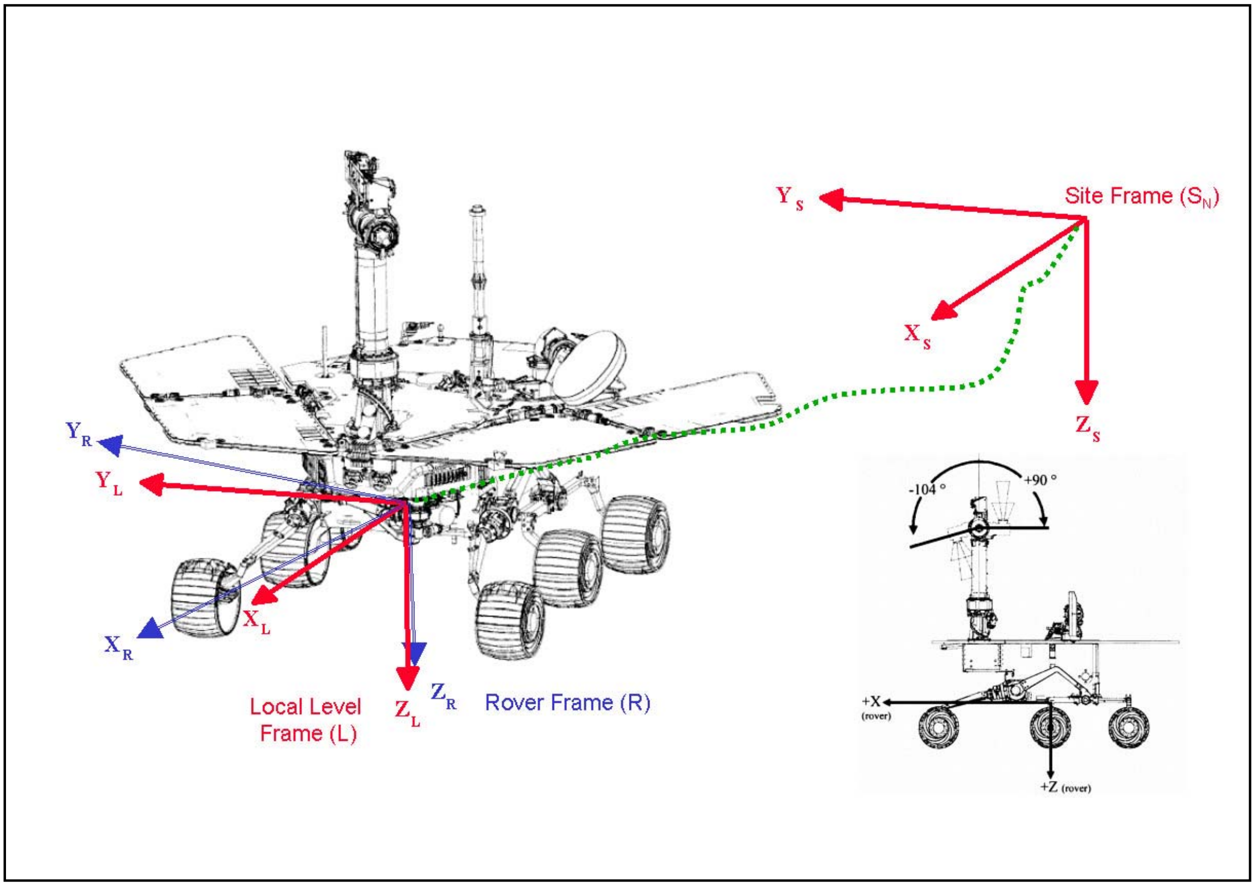

Figure 1: S, S_R, and R Frame Coordinate Systems

2.4.4 Data Storage Conventions

The MB EDR data files contain binary data. Mössbauer and energy spectra are 24-bit integers stored in LSB first order. The drive error signal data are 16-bit integers stored in LSB first order. Temperature data are 16-bit integers stored in MSB first order. The detached PDS labels for MB 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 after data has been received, 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 being. Validation in this case will include inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the target shown / apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure.

3. DETAILED DATA PRODUCT SPECIFICATIONS

3.1 Data Product Structure and Organization

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

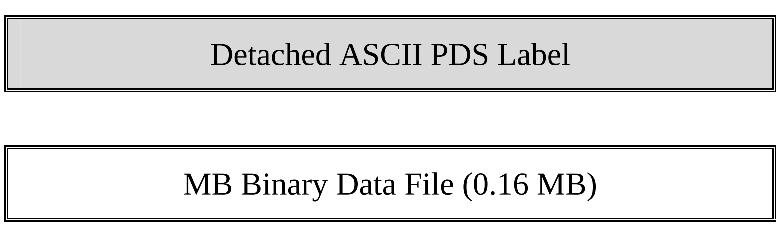

Figure 2: The MB EDR consists of two files.

3.2 Data Format Descriptions

A MB EDR data product consists of five contiguous blocks of binary data or a single block and a detached ASCII PDS label (see section 3.3.1). Each data block is 32 Kbytes long for a total size of 160 Kbytes or 32 Kbytes. The five data blocks are a copy of the instrument's memory buffer (Figure 3). The EDR file contains Mössbauer spectra for a given target measured in thirteen temperature windows, along with temperature data, an energy spectrum, a drive error signal spectrum, copies of the instrument parameter block and logbook. In addition, the file contains ten compressed Mössbauer spectra stored in the EEPROM as backup in case the instrument loses power. These backup spectra are updated during data acquisition, approximately every 6 minutes. The MB spectrum stored at location 5400h-71FFh in the fifth block is for backup (a copy of data from one of main temperature windows) and contains no new data.

Block Number	Contents
1	SRAM (0000h – 7FFFh of bank 0)
2	SRAM (8000h – FFFFh of bank 0)
3	SRAM (0000h – 7FFFh of bank 1)
4	SRAM (8000h – FFFFh of bank 1)
5	0000h – 17FFh FRAM 1800h – 53FFh EEPROM backup 5400h – 71FFh One Mössbauer temperature window 7200h – 75FFh Drive error signal 7600h – 77FFh Parameter block from SRAM 7800h – 7DFFh Temperature data 7E00h – 7FF5h Spare bytes 7FF6h – 7FFFh Hardware ID

Figure 3: Data structure for a MB binary file.

Data from the instrument's SRAM occupy the first four blocks in the Mössbauer EDR data product. The SRAM structure is shown in Figure 4. It contains three copies of the instrument parameter block, a drive error signal, temperature data, an energy spectrum, and Mössbauer spectra from 13 temperature windows. Sections of the SRAM that are not described in Figure 4 contain internal instrument variables and temporary data, which should be considered as spare bytes.

Bank 0		Bank 1	
0000h-05FFh	Three copies of instrument parameters (512 byte each)		
0654h-0A53h	Drive error signal (1024 bytes)		
1100h-16FFh	Temperature data (1536 bytes)		
1F00h-2DFFh	Energy spectrum (3840 bytes)		
2E00h-E1FFh	Six MB temperature windows (temp. windows 8-13, 7680 bytes each)	1000h-E1FFh	Seven MB temperature windows (temp. windows 1-7, 7680 bytes each)

Figure 4: SRAM structure (only scientific data shown).

Each copy of the instrument parameter block in the SRAM is 512 bytes long. Table 4 lists the contents of the instrument parameter block. The SRAM has three copies of the parameter block for a total of 1536 bytes. The drive error signal, which monitors the operation of the MB drive unit, has 512 channels with two bytes per channel. The drive error signal values are stored in LSB first order. The MB EDR file contains temperature data recorded by three sensors: one on the electronics board (located inside the rover's Warm Electronics Box), one on the sensor head, and one on the reference sensor. Each temperature value is scaled and stored as a two byte integer in MSB first order. The file contains 256 temperature records. Within each record, the order of values is board sensor, sample sensor, and reference sensor. The following equations contain the conversion of the scaled temperatures into Kelvin:

Board sensor = 273.2 + 25 + (scaled value * 1.638*2500/4096 - 608)/2

Sample sensor = (scaled value)/10

Reference sensor = (scaled value)10

The energy spectrum, which monitors the operation of the detectors, contains 256 channels of data for each of 5 detectors. Each value is a three byte integer stored in LSB first order. The data are aggregated so that data for the first detector are followed by data for the second detector and so on. The instrument has four counters to store eight measurements from the detectors (4 detectors with 2 energies each). As a result, the measurements from pairs of detectors are added together and stored in one counter. These four summed spectra are included in the EDR data products. In the generation of the Reduced Data Record (RDR) products, the two 6.4 keV spectra are summed, as are the two 14.4 keV. These summed spectra exist, and are labeled as separate products in the RDR data set.

Mössbauer data collected during a measurement are divided into temperature windows defined by a selectable lookup table. Each Mössbauer temperature window consists of five spectra (one for each detector). One spectrum contains 512 channels with three bytes per channel stored in LSB first order. The first channel of each spectrum gives the measurement lifetime, where lifetime is the number of cycles that the MB drive unit has executed during the measurement. The collection of MB spectra can be described as a three-dimensional array with axes of temperature window, detector, and channel, with the channel axis varying the fastest.

The fifth data block of the MB EDR data file contains a copy of the instrument FRAM and EEPROM memory, along with copies of one Mössbauer temperature window, the drive error signal, instrument parameter block, and temperature data. The FRAM consists of three copies of the instrument parameter block (Table 4) and the instrument logbook. The logbook has 256 records with eight bytes in each record. The logbook contains information on the engineering status of the instrument during a measurement. The information in the logbook is not needed and not useful for calibration or science data analysis. The EEPROM contains ten compressed MB spectra. The data are compressed by combining data from several temperature windows or from several detectors. Each compressed MB spectrum has 512 channels with three bytes per channel stored in LSB first order.

The integration times for each MB measurement are listed in the observation table provided to the PDS. This table can be found in the document directory of both the EDR and RDR data sets. Integration time is calculated by division of drive cycles by drive frequency. As noted, the number of drive cycles is included with each spectrum in the EDR data files. The drive frequency is about 25 Hz. It can also be derived from the FG_PRESCALER parameter stored in the instrument parameter block. The stored FG_PRESCALER value is converted to drive frequency as follows: fdrive = 900 Hz / FG_PRESCALER. So, for example, a prescaler value of 37 yields a drive frequency of 900/37 = 24.3 Hz.

Table 4: MB Instrument Parameters

Parameter name	Address in FRAM (decimal)	Size (bytes)	Description
DEFAULT_MODE	00	1	Mode after hard reset
CURRENT_MODE	01	1	Current mode of the instrument
COUNTER_CONTROL	02	4	Counters configuration
MAX_VELOCITY	06	2	Maximum velocity of MB drive
FG_PRESCALER	08	1	Sets drive frequency
WINDOW_WIDTH	12	1	Width (in DAC-channels) for one energy spectra channel
ESP_ACQ_TIME	13	1	Acquisition time for one energy channel measured in MB-cycles
DIFFSIG_ACQ_TIME	15	1	Acquisition time for drive error signal, measured in MB-cycles
TEMPER_CYCLE	21	1	Interval (in 256 MB cycles) between temperature measurements
TEST_MODUS	26	1	255 -> test mode (increment), else counters count
BACKUP_CYCLE	27	1	Interval (in 256 MB cycles) between backup updating
TEMPER_WIN_SAVE	34	1	Number of MB temperature window to be saved in last data block

THRESHOLDS	50	288	Detector thresholds (18 thresholds x 2 bytes x 8 temperature windows)
TEMP_THRES	338	26	Temperature thresholds for Mössbauer spectra
TEMP_THRES_DET	364	18	Temperature thresholds for detector thresholds
BACKUP_TARGET	396	70	Backup target of each of the 65 spectra

3.3 Label and Header Descriptions

3.3.1 PDS Label

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

PDS labels are written in Object Description Language (ODL) [7]. PDS label statements have the form of "keyword = value". Each label statement is terminated with a carriage return character (ASCII 13) and a line feed character (ASCII 10) sequence to allow the label to be read by many operating systems. Pointer statements with the following format are used to indicate the location of data objects in the file:

^object = location

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

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

3.3.1 PDS Data Objects

The MB EDR consists of a heterogeneous set of data structures. The PDS label that describes these structures uses several low-level data objects. The PDS collection data object is used for the file as a whole. Within the collection object, the MB data structures are defined by PDS array and element data objects. The PDS array data object is used for multi-dimensional arrays of homogeneous objects. The array object is used to describe the Mössbauer spectra, energy spectrum, drive error signal, temperature data, and the instrument parameter block and logbook. The AXES keyword within the array object gives the dimension of the array, whereas the AXIS_ITEMS keyword gives the number of elements for each axis. The data are stored so that the rightmost axis is the fastest varying axis. The PDS element data object is used to describe the individual constituents of an array. The spare bytes and hardware ID fields within the MB EDR data file are also described with the PDS element data object. The PDS collection, array, and element data objects are further defined in the PDS standard reference [7].

4. APPLICABLE SOFTWARE

4.1 Utility Programs

A reader program, called mb2asc, will be available for a variety of computer platforms to convert the MB 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 A MÖSSBAUER EDR LABEL CONTAINING ALL 5 BLOCKS

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 32768
FILE_RECORDS                    = 5
^COLLECTION                     = "1B123456789EDR0205C0062N0M1.DAT"

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MER1-M-MB-2-EDR-V1.0"
PRODUCT_ID                      = "1B123456789EDR0205C0062N0M1”
PRODUCT_TYPE                    = "MB_EDR"
PRODUCT_VERSION_ID              = "V2.0 D-22849"
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                   = MB
INSTRUMENT_TYPE                 = SPECTROMETER
INSTRUMENT_VERSION_ID           = <FM1, FM2, “UNK”>
LOCAL_TRUE_SOLAR_TIME           = “09:16:03”
MAGNET_ID                       = ”NULL”
MISSION_NAME                    = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME              = <"PRIMARY MISSION", TBD>
OBSERVATION_ID                  = “0”
PLANET_DAY_NUMBER               = 3
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          = 33
APPLICATION_PROCESS_NAME        = MB
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              = ALL
TELEMETRY_PROVIDER_ID            = "SSW MER_DP"
TELEMETRY_SOURCE_NAME            = “Input DP Filename”
TELEMETRY_SOURCE_TYPE            = “DATA PRODUCT”
TLM_INST_DATA_HEADER_ID          = 1

/* 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"

/* INSTRUMENT DATA ELEMENTS */
/* 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 = MB
END_GROUP                         = START_IDD_ARTICULATION_STATE

/* COORDINATE SYSTEM STATE: INSTRUMENT_DEPLOYMENT DEVICE AT THE START*/

GROUP                             = START_IDD_COORDINATE_SYSTEM
  COORDINATE_SYSTEM_NAME          = MB_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 THE 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        = “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 = MB
END_GROUP                        = STOP_IDD_ARTICULATION_STATE

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

GROUP                            = STOP_IDD_COORDINATE_SYSTEM
  COORDINATE_SYSTEM_NAME         = MB_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                           = COLLECTION
  NAME                           = MOESSBAUER_DATA_FILE
  BYTES                          = 163840
  INTERCHANGE_FORMAT             = BINARY
  DESCRIPTION                    = "Binary data structure containing
      MB measurements"

  OBJECT                         = ARRAY
    NAME                         = INSTR_PARAM_1
    AXES                         = 2
    AXIS_ITEMS                   = (3,512)
    START_BYTE                   = 1
    BYTES                        = 1536
    DESCRIPTION                  = "Three copies of instrument
      parameters (512 bytes each)."
  END_OBJECT                     = ARRAY

  OBJECT                         = ELEMENT
    NAME                         = SPARE_01
    DATA_TYPE                    = UNSIGNED_INTEGER
    START_BYTE                   = 1537
    BYTES                        = 84
    DESCRIPTION                  = "Unused bytes."
  END_OBJECT                     = ELEMENT

  OBJECT                         = ARRAY
    NAME                         = DRIVE_ERROR_SIGNAL_1
    AXES                         = 1
    AXIS_ITEMS                   = 512
    START_BYTE                   = 1621
    DESCRIPTION                  = "The drive error signal has
      values for 512 channels with 2 bytes per channel in LSB
      first order for a total of 1024 bytes."
    OBJECT                       = ELEMENT
```

```
      NAME                        = CHANNEL
      DATA_TYPE                   = LSB_INTEGER
      BYTES                       = 2
    END_OBJECT                    = ELEMENT
  END_OBJECT                      = ARRAY

  OBJECT                          = ELEMENT
    NAME                          = SPARE_02
    DATA_TYPE                     = UNSIGNED_INTEGER
    START_BYTE                    = 2645
    BYTES                         = 1708
    DESCRIPTION                   = "Unused bytes."
  END_OBJECT                      = ELEMENT

  OBJECT                          = ARRAY
    NAME                          = TEMPERATURE_1
    AXES                          = 2
    AXIS_ITEMS                    = (256,3)
    AXIS_NAME                     = ("TIME", "SENSOR")
    START_BYTE                    = 4353
    DESCRIPTION                   = "Temperature values for sensors on
      electronics board, sample, and reference. Each value is 2
      bytes (MSB first). There are 256 for each sensor for a total
      of 1536 bytes."
    OBJECT                        = ELEMENT
      NAME                        = TEMPERATURE
      DATA_TYPE                   = MSB_INTEGER
      BYTES                       = 2
    END_OBJECT                    = ELEMENT
  END_OBJECT                      = ARRAY

  OBJECT                          = ELEMENT
    NAME                          = SPARE_03
    DATA_TYPE                     = UNSIGNED_INTEGER
    START_BYTE                    = 5889
    BYTES                         = 2048
    DESCRIPTION                   = "Unused bytes."
  END_OBJECT                      = ELEMENT

  OBJECT                          = ARRAY
    NAME                          = ENERGY_SPECTRA_1
    AXES                            = 2
    AXIS_ITEMS                    = (5,256)
    AXIS_NAME                     = ("DETECTOR", "CHANNEL")
    START_BYTE                    = 7937
    DESCRIPTION                   = "Energy spectra for each of the
      5 detectors with 256 channels per detector. Each value is
      3 bytes long (LSB first) for a total of 3840 bytes."
    OBJECT                        = ELEMENT
      NAME                        = COUNTS
      DATA_TYPE                   = LSB_INTEGER
      BYTES                       = 3
    END_OBJECT                    = ELEMENT
  END_OBJECT                      = ARRAY

  OBJECT                          = ARRAY
    NAME                          = MOESSBAUER_SPECTRA_1
```

```
    AXES                          = 3
    AXIS_ITEMS                    = (6,5,512)
    AXIS_NAME                     = ("TEMPERATURE WINDOW",
                                     "DETECTOR","CHANNEL")
    START_BYTE                    = 11777
    DESCRIPTION                   = "Moessbauer spectra for each of
      the 5 detectors with 512 channels per detector. There are
      six temperature windows (numbers 8 to 13) in this part of the
      file. The first channel of each spectrum contains the lifetime
      for the spectrum. Each value is 3 bytes long (LSB first) for
      a total of 46080 bytes."
    OBJECT                        = ELEMENT
      NAME                        = COUNTS
      DATA_TYPE                   = LSB_INTEGER
      BYTES                       = 3
    END_OBJECT                    = ELEMENT
  END_OBJECT                      = ARRAY

  OBJECT                          = ELEMENT
    NAME                          = SPARE_04
    DATA_TYPE                     = UNSIGNED_INTEGER
    START_BYTE                    = 57857
    BYTES                         = 11776
    DESCRIPTION                   = "Unused bytes."
  END_OBJECT                      = ELEMENT

  OBJECT                          = ARRAY
    NAME                          = MOESSBAUER_SPECTRA_2
    AXES                          = 3
    AXIS_ITEMS                    = (7,5,512)
    AXIS_NAME                     = ("TEMPERATURE WINDOW",
                                     "DETECTOR","CHANNEL")
    START_BYTE                    = 69633
    DESCRIPTION                   = "Moessbauer spectra for each of
      the 5 detectors with 512 channels per detector. There are
      seven temperature windows (numbers 1 to 7) in this part of the
      file. The first channel of each spectrum contains the lifetime
      for the spectrum. Each value is 3 bytes long (LSB first) for
      a total of 53760 bytes."
    OBJECT                        = ELEMENT
      NAME                        = COUNTS
      DATA_TYPE                   = LSB_INTEGER
      BYTES                       = 3
    END_OBJECT                    = ELEMENT
  END_OBJECT                      = ARRAY

  OBJECT                          = ELEMENT
    NAME                          = SPARE_05
    DATA_TYPE                     = UNSIGNED_INTEGER
    START_BYTE                    = 123393
    BYTES                         = 7680
    DESCRIPTION                   = "Unused bytes."
  END_OBJECT                      = ELEMENT

  OBJECT                          = COLLECTION
    NAME                          = FRAM
    BYTES                         = 6144
```

```
    START_BYTE                  = 131073
    DESCRIPTION                 = "Copy of the instrument FRAM
      memory (6144 bytes total)."
    OBJECT                      = ARRAY
      NAME                      = INSTR_PARAM_2
      AXES                      = 2
      AXIS_ITEMS                = (3,512)
      START_BYTE                = 1
      BYTES                     = 1536
      DESCRIPTION               = "Three copies of instrument
        parameters (512 bytes each)."
    END_OBJECT                  = ARRAY
    OBJECT                      = ARRAY
      NAME                      = LOGBOOK
      AXES                      = 1
      AXIS_ITEMS                = 256
      START_BYTE                = 1537
      DESCRIPTION               = "Instrument logbook, which has
        256 entries. Each entry is 8 bytes (2048 bytes total)."
      OBJECT                    = ELEMENT
        NAME                    = LOGBOOK_ENTRY
        DATA_TYPE               = UNSIGNED_INTEGER
        BYTES                   = 8
      END_OBJECT                = ELEMENT
    END_OBJECT                  = ARRAY
    OBJECT                      = ELEMENT
      NAME                      = SPARE_06
      DATA_TYPE                 = UNSIGNED_INTEGER
      START_BYTE                = 3585
      BYTES                     = 2560
      DESCRIPTION               = "Unused bytes."
    END_OBJECT                  = ELEMENT
  END_OBJECT                    = COLLECTION

  OBJECT                        = ARRAY
    NAME                        = COMPRESSED_SPECTRA
    AXES                        = 2
    AXIS_ITEMS                  = (10,512)
    AXIS_NAME                   = ("SPECTRUM","CHANNEL")
    START_BYTE                  = 137217
    DESCRIPTION                 = "Compressed Moessbauer spectra as
      backup to data in SRAM. There are 10 spectra with 512 channels
      per spectrum. Each value is 3 bytes long (LSB first) for a
      total of 15360 bytes."
    OBJECT                      = ELEMENT
      NAME                      = COUNTS
      DATA_TYPE                 = LSB_INTEGER
      BYTES                     = 3
    END_OBJECT                  = ELEMENT
  END_OBJECT                    = ARRAY

  OBJECT                        = ARRAY
    NAME                        = MOESSBAUER_SPECTRA_3
    AXES                        = 1
    AXIS_ITEMS                  = (5,512)
    AXIS_NAME                   = ("DETECTOR","CHANNEL")
    START_BYTE                  = 152577
```

```
    DESCRIPTION                  = "Moessbauer spectra for each of
      the 5 detectors with 512 channels per detector. There is
      one temperature window in this part of the file. The first
      channel contains the lifetime for the spectrum. Each value
      is 3 bytes long (LSB first) for a total of 7680 bytes."
    OBJECT                       = ELEMENT
      NAME                       = COUNTS
      DATA_TYPE                  = LSB_INTEGER
      BYTES                      = 3
    END_OBJECT                   = ELEMENT
  END_OBJECT                     = ARRAY

  OBJECT                         = ARRAY
    NAME                         = DRIVE_ERROR_SIGNAL_2
    AXES                         = 1
    AXIS_ITEMS                   = 512
    START_BYTE                   = 160257
    DESCRIPTION                  = "The drive error signal has
      values for 512 channels with 2 bytes per channel in LSB
      first order for a total of 1024 bytes."
    OBJECT                       = ELEMENT
      NAME                       = CHANNEL
      DATA_TYPE                  = LSB_INTEGER
      BYTES                      = 2
    END_OBJECT                   = ELEMENT
  END_OBJECT                     = ARRAY

  OBJECT                         = ARRAY
    NAME                         = INSTR_PARAM_3
    AXES                         = 1
    AXIS_ITEMS                   = 512
    START_BYTE                   = 161281
    BYTES                        = 512
    DESCRIPTION                  = "One copy of instrument
      parameters (512 bytes total)."
  END_OBJECT                     = ARRAY

  OBJECT                         = ARRAY
    NAME                         = TEMPERATURE_2
    AXES                         = 2
    AXIS_ITEMS                   = (256,3)
    AXIS_NAME                    = ("TIME","SENSOR")
    START_BYTE                   = 161793
    DESCRIPTION                  = "Temperature values for sensors on
      electronics board, sample, and reference. Each value is 2
      bytes (MSB first). There are 256 for each sensor for a total
      of 1536 bytes."
    OBJECT                       = ELEMENT
      NAME                       = TEMPERATURE
      DATA_TYPE                  = MSB_INTEGER
      BYTES                      = 2
    END_OBJECT                   = ELEMENT
  END_OBJECT                     = ARRAY

  OBJECT                         = ELEMENT
    NAME                         = SPARE_07
    DATA_TYPE                    = UNSIGNED_INTEGER
```

```
    START_BYTE                      = 163329
    BYTES                           = 502
    DESCRIPTION                     = "Unused bytes."
  END_OBJECT                        = ELEMENT

  OBJECT                            = ELEMENT
    NAME                            = HARDWARE_ID
    DATA_TYPE                       = UNSIGNED_INTEGER
    START_BYTE                      = 163831
    BYTES                           = 10
  END_OBJECT                        = ELEMENT
END_OBJECT                          = COLLECTION
END
```

APPENDIX B - EXAMPLE OF A MÖSSBAUER EDR LABEL, SINGLE BLOCK

```
PDS_VERSION_ID                    = PDS3

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                       = "MER1-M-MB-2-EDR-V1.0"
PRODUCT_ID                        = "1B123456789EDR0205C0062N0M1”
PRODUCT_TYPE                      = "MB_EDR"
PRODUCT_VERSION_ID                = "V2.0 D-22849"
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                     = MB
INSTRUMENT_TYPE                   = SPECTROMETER
INSTRUMENT_VERSION_ID             = <FM1, FM2, “UNK”>
LOCAL_TRUE_SOLAR_TIME             = “09:16:03”
MAGNET_ID                         = “NULL”
MISSION_NAME                      = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME                = "PRIMARY MISSION"
OBSERVATION_ID                    = “0”
PLANET_DAY_NUMBER                 = 3
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            = 33
APPLICATION_PROCESS_NAME          = MB
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               = ALL
TELEMETRY_PROVIDER_ID             = "SSW MER_DP"
TELEMETRY_SOURCE_NAME             = “Input DP Filename”
TELEMETRY_SOURCE_TYPE             = “DATA PRODUCT”
```

```
TLM_INST_DATA_HEADER_ID          = 1

/* 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"

/* INSTRUMENT DATA ELEMENTS */
/* 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 = MB
END_GROUP                        = START_IDD_ARTICULATION_STATE

/* COORDINATE SYSTEM STATE: INSTRUMENT_DEPLOYMENT DEVICE AT THE START*/

GROUP                            = START_IDD_COORDINATE_SYSTEM
  COORDINATE_SYSTEM_NAME         = MB_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 THE 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       = “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 = MB
END_GROUP                        = STOP_IDD_ARTICULATION_STATE

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

```
GROUP                               = STOP_IDD_COORDINATE_SYSTEM
  COORDINATE_SYSTEM_NAME            = MB_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

/* FILE DATA ELEMENTS */
OBJECT                              = FILE
  FILE_NAME                         = "1B123456789EDR0205C0062N0M1.DAT"
  RECORD_TYPE                       = FIXED_LENGTH
  RECORD_BYTES                      = 32768
  FILE_RECORDS                      = 1
  SEQUENCE_NUMBER                   = n
  DESCRIPTION                       = "Single Moessbauer 32kB data block. The
    keyword sequence_number indicates which of the five possible data blocks
    is contained in this file. See the EDR data product SIS for details on
    the content of this data block."
END_OBJECT                          = FILE
END
```

APPENDIX C – PDS LABEL ITEMS

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
APPLICATION_PROCESS_ID	Identifies the source/process which created the data.	integer		(see APPLICATION_PROCESS_NAME)	**LOCATION** TELEMETRY (Class) **SOURCE** CCSDS:Primary:APID
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"	**LOCATION** TELEMETRY (Class) **SOURCE** Table Lookup CCSDS:Primary:APID
APPLICATION_PROCESS_SUBTYPE_ID	Identifies the source/subprocess which created the data.	integer		0 MB SCIENCE DATA of one or five blocks of data	**LOCATION** TELEMETRY (Class) **SOURCE** CCSDS:Primary:subtype

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
ARTICULATION_DEVICE_ANGLE	Provides the value of an angle between two parts or segments of an articulated device. Note: MER uses radians. The PDS default unit for this keyword is degrees, so the <rad> tag is required for MER data.	float array[10]	radians (<rad> unit tag required)		**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** Group Dependent a) IDPH:bgn->idd.q_enc[5] IDPH:bgn->idd.q_pot[5] b) IDPH:end->idd.q_enc[5] IDPH:end->idd.q_pot[5]
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")	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) Static Values b) Static Values
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"	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) Static Values b) Static Values

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
ARTICULATION_DEVICE_MODE	Indicates the deployment state (i.e., physical configuration) of an articulation device at the time of data acquisition. For MER, this is the mode of the last move: FREE SPACE-IDD arm was moved where there was no contact with a target expected. GUARDED-IDD arm was moved where contact with the target was expected. RETRACTING-IDD arm was moved where an instrument is removed from a target. PRELOAD-IDD arm stays in contact with the target and applies force or overtravel on an instrument.	string		IDD 0="FREE SPACE", 1="GUARDED", 2="RETRACTING", 3="PRELOAD"	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** Group Dependent; Table Lookup a) IDPH:bgn->idd.idd_mode b) IDPH:end->idd.idd_mode
ARTICULATION_DEVICE_NAME	Specifies the common name of the articulation device described by the containing group. Note: The associated ARTICULATION_DEVICE_ID element provides an abbreviated name or acronym for the articulated device.	string		"INSTRUMENT DEPLOYMENT DEVICE"	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) Static Value b) Static Value
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	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) IDPH:bgn->idd.temp[2] b) IDPH:end->idd.temp[2]
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")	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) Static Values b) Static Values

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
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>** a) Static Value b) 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>** Lookup Table a) IDPH:bgn->idd.contact (bit map) b) IDPH:end->idd.contact (bit map)

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
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")	**LOCATION** a) START_IDD_ARTICULATION_STATE (Group) b) STOP_IDD_ARTICULATION_STATE (Group) **SOURCE** a) Static Values b) Static Values
COORDINATE_SYSTEM_INDEX	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. 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. 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 coordinate system instances. It is legal to use any number of indices to describe a coordinate system instance, however. Example: COORDINATE_SYSTEM_INDEX = (1,3,2,3,2)	integer array[5]			**LOCATION** a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** Calculated from COORDINATE_SYSTEM_NAME and the rmc value. Use all 5 elements except if the coordinate_SYSTEM_NAME = SITE_FRAME, then only use the first element.

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
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")	**LOCATION** a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** Static Value; Group Dependant Should match the number of values in the COORDINATE_SYSTEM_INDEX
COORDINATE_SYSTEM_NAME	Provides the full name of the coordinate system to which the state vectors are referenced. 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.	string(30)		"ROVER_FRAME", "SITE_FRAME", "MAST_FRAME", "LOCAL_LEVEL_FRAME", IDD only 0="MI_FRAME" 1="RAT_FRAME" 2="MB_FRAME" 3="APXS_FRAME",	**LOCATION** a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** Group dependant a) one of the IDD only frames based on IDPH:bgn->idd.instrument b) one of the IDD only frames based on IDPH:end->idd.instrument c) ROVER_FRAME
DATA_SET_ID	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. 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.	string(40)		"MER1-M-MB-2-EDR-OPS-V1.0", "MER2-M-MB-2-EDR-OPS-V1.0" "SIM1-M-MB-2-EDR-V1.0", "SIM2-M-MB-2-EDR-V1.0"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Table Lookup CCSDS:Primary:APID CHDO-82:scft_id
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]	**LOCATION** TELEMETRY (Class) **SOURCE** Calculated CHDO_82:ert

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
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]	**LOCATION** TELEMETRY (Class) **SOURCE** Calculated CHDO_82:ert
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			**LOCATION** TELEMETRY (Class) **SOURCE** UPTH:TotalParts
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		1 or 5	**LOCATION** FILE DATA ELEMENT (Class) **SOURCE** Static Value
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)		**SCID** **Keyword Values** 253 "MER1" 254 "MER2" 253 "SIM1" 255 "SIM2"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Table Lookup CHDO_82:scft_id
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		"MARS EXPLORATION ROVER 1" "MARS EXPLORATION ROVER 2" "SIMULATED MARS EXPLORATION ROVER 1" "SIMULATED MARS EXPLORATION ROVER 2"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Table Lookup CHDO_82:scft_id

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
INSTRUMENT_ID	Provides an abbreviated name or acronym which identifies an instrument. Note: INSTRUMENT_ID is not a unique identifier for a given instrument. Note also that the associated INSTRUMENT_NAME element provides the full name of the instrument. Example values: IRTM (for Viking Infrared Thermal Mapper), PWS (for plasma wave spectrometer).	string(12)		IDD 29 = "MI" 34 = "RAT" 33 = "MB" 32 = "APXS"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Table Lookup; CCSDS:Primary:APID
INSTRUMENT_TYPE	Identifies the type of an instrument. Example values: POLARIMETER, RADIOMETER, REFLECTANCE SPECTROMETER, VIDICON CAMERA. Note that mosaics may contain more than one value in an array.	string		“SPECTROMETER”	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value
INSTRUMENT_VERSION_ID	Identifies the specific model of an instrument used to obtain data. For example, this keyword could be used to distinguish between an engineering model of a camera used to acquire test data, and a flight model of a camera used to acquire science data during a mission.	string(8)		“FM1”, “FM2”, “QM”	**LOCATION** IDENTIFICATION (Class) **SOURCE** Table lookup Only if block 5 exist

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
LOCAL_TRUE_SOLAR_TIME	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. The coordinate system used to define LTST has its origin at the center of the planet. Its Z- axis is the north pole vector (or spin axis) of the planet. The X- axis is chosen to point in the direction of the vernal equinox of the planet' s orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet' s center to the Sun is perpendicular to the planet' s north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet' s equator.) Positions of points in this frame can be expressed as a radius and areocentric ' right ascension' and ' declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet' s surface. Specifically, LTST = (a(P) – a(TS)) * (24 / 360) + 12 where, LTST = the local true solar time in true solar hours a(P) = ARA of the point on the planet' s surface in deg a(TS) = ARA of the true sun in deg	string(12)		NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	<u>LOCATION</u> •IDENTIFICATION (Class) <u>SOURCE</u> •Calculation: - IDPH:bgn->rover_time.seconds - IDPH:bgn->rover_time.subseconds - SCLK Kernel - Landing- Site kernel - P Kernel

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
MAGNET_ID	Identifies a magnet instrument that is visible in an image or observation.	string		"N/A", "CAPTURE", "FILTER", "UNK","NULL"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated
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"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value
MISSION_PHASE_NAME	Provides the commonly-used identifier of a mission phase.	string(30)		" "PRIMARY MISSION", "TBD"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Operator Supplied Parameter
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			**LOCATION** IDENTIFICATION (Class) **SOURCE** IPDH:bgn->context_id
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 fo the PMA (camera head) boresight (about the ORIGIN_OFFSET_VECTOR) relative to the Rover frame.	float array[3]	meters		**LOCATION** a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** a) IDPH:bgn->idd.pos[3] b) IDPH:end->idd.pos[3 c) IDPH:bgn->pos.v[3]

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
ORIGIN_ROTATION_QUATERNION	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. Quaternions are expressed as a set of four numbers in the order (s, v1, v2, v3), where s = cos(theta/2) v(n) = sin(theta/2)*a(n). theta = the angle of rotation a = the (x,y,z) vector around which the rotation occurs. 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 POSITVE_ELEVATION_DIRECTION is declared to be "UP" or "DOWN" 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[4]			**<u>LOCATION</u>** a) START_IDD_COORDINATE_SYSTEM (Group) b) STOP_IDD_COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **<u>SOURCE</u>** a) IDPH:bgn->idd.quaternion[4] b) IDPH:end->idd.quaternion[4] c) IDPH:bgn->pos.q[4] Received in (v1, v2, v3, s0 order.

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
PACKET_MAP_MASK	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. The number is stored in the PDS radix notation of <radix>#<value>#.	non-decimal			**LOCATION** TELEMETRY (Class) **SOURCE** Calculated UPDTH:PartNumber
PDS_VERSION_ID	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'. Examples: PDS3, PDS4.	string(6)		"PDS3"	**LOCATION** PDS required **SOURCE** PDS
PLANET_DAY_NUMBER	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).	integer		NOTE: Value will be uncalibrated if SPICE kernels are not available.	**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated IDPH:bgn->rovert_time.seconds IDPH:bgn->rover_time.subseconds SCLK kernel
POSITIVE_AZIMUTH_DIRECTION	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.	string		"CLOCKWISE", "COUNTERCLOCKWISE"	**LOCATION** a) ROVER_COORDINATE_SYSTEM (Group) b) START_IDD_COORDINATE_SYSTEM (Group) c) STOP_IDD_COORDINATE_SYSTEM (Group) **SOURCE** a) Static Value b) Static Value c) Static Value

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
POSITIVE_ELEVATION_DIRECTION	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. A value of UP indicates that elevation is measured positively upwards, i.e., the zenith point would be at +90 degrees and the nadir point at -90 degrees. DOWN indicates that the elevation is measured positively downwards; the zenith point would be at -90 degrees and the nadir point at +90 degrees. For MER, which follows the Mars Pathfinder convention, increasing elevation ("UP") moves towards the negative Z axis.	string		"UP", "DOWN"	**LOCATION** a) ROVER_COORDINATE_SYSTEM (Group) b) START_IDD_COORDINATE_SYSTEM (Group) c) STOP_IDD_COORDINATE_SYSTEM (Group) **SOURCE** a) Static Value b) Static Value c) Static Value
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"	**LOCATION** HISTORY (Class) **SOURCE** Static Value
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"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value
PRODUCT_CREATION_TIME	Defines the UTC system format time when a product was created. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]Z	string			**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated
PRODUCT_ID	Represents a permanent, unique identifier assigned to a data product by its producer. See also: source_product_id. Note: In the PDS, the value assigned to product_id must be unique within its data set. Additional note: The product_id can describe the lowest-level data object that has a PDS label.	string(40)		Filename less the extension	**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated
PRODUCT_TYPE	Identifies the type or category of a data product within a data set.	string(8)		"MB_EDR"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
PRODUCT_VERSION_ID	Identifies the version of an individual product within a data set. PRODUCT_VERSION_ID is intended for use within AMMOS to identify separate iterations of a given product, which will also have a unique FILE_NAME. For MER, PRODUCT_VERSION_ID includes a Version field that begins with “V” followed by the Version decimal number of the controlling SIS document. Example: “V2.0 D-22849” Note: This might not be the same as the data set version that is an element of the DATA_SET_ID value.	string(12)		“V<vernum> D-22849””	**LOCATION** IDENTIFICATION (Class) **SOURCE** User Parameter
QUATERNION_MEASUREMENT_METHOD	Specifies the quality of the rover orientation estimate. “UNKNOWN” - The attitude should simply not be trusted. This is the initial grade given on Landing, for example. “TILT_ONLY” - The attitude estimate is only good for tilt determination (2-axis knowledge). Activities which require azimuth knowledge should be careful. “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”). “FINE” - Sungaze completed successfully, and the attitude estimate is sufficient for pointing HGA (a.k.a. “ThreeAxisFine”).	string		0 = “UNKNOWN” 1 = “TILT_ONLY” 2 = “COARSE” 3 = “FINE”	**LOCATION** ROVER_COORDINATE_SYSTEM (Group) **SOURCE** Table Lookup IDPH:bgn->rvr_quat_qlty

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
RECEIVED_PACKETS	Provides the total number of telemetry packets which constitute a reconstructed data product.	integer			**LOCATION** TELEMETRY (Class) **SOURCE** Calculated UPTH:PacketNumber
RECORD_BYTES	Indicates the number of bytes in a physical file record, including record terminators and separators. Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.	integer		0 to n	**LOCATION** FILE (Class) **SOURCE** Calculated
RECORD_TYPE	Indicates the record format of a file. Note: In the PDS, when record_type is used in a detached label file it always describes its corresponding detached data file, not the label file itself. The use of record_type along with other file-related data elements is fully described in the PDS Standards Reference.	string(20)		"FIXED_LENGTH"	**LOCATION** FILE (Class) **SOURCE** Static Value

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
REFERENCE_COORD_SYSTEM_INDEX	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. 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 first two 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. See also REFERENCE_COORD_SYSTEM_NAME and COORDINATE_SYSTEM_INDEX.	integer array[5]			**LOCATION** a) START_IDD_ COORDINATE_SYSTEM (Group) b) STOP_IDD_ COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** # of values used is group dependent a) IDPH:bgn->rmc(0,1) b) IDPH:bgn->rmc(0,1) c) IDPH:bgn->rmc√(0) Calculated from IDPH:bgn->rmc and COORDINATE_SYSTEM_NAME Use all five elements except if the REFERENCE_COORD_SYSTEM_NAME = SITE_FRAME, then only use the first element.
REFERENCE_COORD_SYSTEM_NAME	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. For MER, the reference is usually a SITE frame.	string(20)		"ROVER_FRAME", "SITE_FRAME", "MAST_FRAME", “LOCAL_LEVEL_FRAME”,	**LOCATION** a) START_IDD_ COORDINATE_SYSTEM (Group) b) STOP_IDD_ COORDINATE_SYSTEM (Group) c) ROVER_COORDINATE_SYSTEM (Group) **SOURCE** # of values used is group dependent a) “ROVER_FRAME” b) “ROVER_FRAME” c) “SITE_FRAME”

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
RELEASE _ID	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” 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” 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”).	string			**LOCATION** IDENTIFICATION **SOURCE** User Parameter

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
ROVER_MOTION_COUNTER	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. 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. 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).	integer array[5]			**LOCATION** IDENTIFICATION (Class) **SOURCE** IDPH:bgn->rmc(5)
ROVER_MOTION_COUNTER_NAME	An array that provides the formal names identifiying each integer in ROVER_MOTION_COUNTER.	string array[5]		("SITE", "DRIVE", "IDD", "PMA", "HGA")	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
SAMPLING_COUNT	The sampling count element provides the number of data samples taken by an instrument or detector.	integer			**LOCATION** TELEMETRY (Class) **SOURCE** IDPH:telemetry_count
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)			**LOCATION** IDENTIFICATION (Class) **SOURCE** UPTH:PrdCmdID:SequenceID
SEQUENCE_VERSION_ID	Provides the version identifier for a particular observation sequence used during planning or data processing.	string(30)			**LOCATION** IDENTIFICATION (Class) **SOURCE** UPTH:PrdCmdID:SequenceVersion
SOFTWARE_NAME	Identifies data processing software such as a program or a program library.	string(60)		"MERTELEMPROC"	**LOCATION** HISTORY (Class) **SOURCE** User Parameter
SOFTWARE_VERSION_ID	Indicates the version (development level) of a program or a program library.	string(20)			**LOCATION** HISTORY (Class) **SOURCE** User Parameter
SPACECRAFT_CLOCK_CNT_PARTITION	Indicates the clock partition active for the SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT elements.	integer		1	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value
SPACECRAFT_CLOCK_START_COUNT	Provides the value of the spacecraft clock at the beginning of a time period of interest. Format is dddddddddd.ddd, measured in units of seconds and stored internally as a floating point number. 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	string(30)			**LOCATION** IDENTIFICATION (Class) **SOURCE** IDPH:bgn->rover_time.seconds IDPH:bgn->rover_time.subseconds Note: It is possible that the sclk can be earlier than the one reported in the DVT time of the data product meta data and the sclk in the data product filename.

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
SPACECRAFT_CLOCK_STOP_COUNT	Provides the value of the spacecraft clock at the end of a time period of interest. Format is ddddddddd.ddd, measured in units of seconds and stored internally as a floating point number.	string(30)			**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated IDPH:end->rover_time.seconds IDPH:end->rover-time.subseconds
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)			**LOCATION** TELEMETRY (Class) **SOURCE** User parameter
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			**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated IDPH:bgn->rover_time.seconds IDPH:bgn->rover_time.subseconds SCLK kernel
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: YYY-MM-DDThh:mm:ss[.fff]	string			**LOCATION** IDENTIFICATION (Class) **SOURCE** Calculated IDPH:end->rover_time.seconds IDPH:end->rover_time.subseconds IDPH:ImgTlmHdr:exp_time SCLK kernel
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"	**LOCATION** IDENTIFICATION (Class) **SOURCE** MARS
TARGET_TYPE	Identifies the type of a named target.	string		"CALIBRATION", "DUST", "N/A", "SUN", "PLANET"	**LOCATION** IDENTIFICATION (Class) **SOURCE** Static Value – PLANET
TELEMETRY_FORMAT_ID	A telemetry format code	string(3)		MB 120=ALL (ALL BLOCKS) 121=BK1 (BLOCK1) 122=BK2 (BLOCK2) 123=BK3 (BLOCK3) 124=BK4 (BLOCK4) 125=BK5 (BLOCK5)	**LOCATION** TELEMETRY (Class) **SOURCE** Table Lookup IDPH: flavor

Keyword Name	Definition	Type	Units	Valid Values	• Location in PDS Label • Source
TELEMETRY_PROVIDER_ID	Identifies the provider and version of the telemetry data used in the generation of this data.	string		"SSW MER_DP" TTACS	**LOCATION** TELEMETRY (Class) **SOURCE** User Parameter
TELEMETRY_SOURCE_NAME	Identifies the name the source of the telemetry source used in the creation of this data set.	string			**LOCATION** TELEMETRY (Class) **SOURCE** Name of the input data product.
TELEMETRY_SOURCE_TYPE	Classifies the source of the telemetry used in creation of this data set.	string(12)		"DATA PRODUCT", SFDU	**LOCATION** TELEMETRY (Class) **SOURCE** User Parameter
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			**LOCATION** TELEMETRY (Class) **SOURCE** IDPH: version