Mars Science Laboratory (MSL)
Software Interface Specification (SIS)
Dynamic Albedo of Neutrons (DAN)
Experiment Data Record (EDR)

DAN EDR SIS Version 1
JPL D-38113
MSL-437-3678

DATE: Jan 31, 2013

Prepared by:
Costin Radulescu, OPGS (JPL)

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

SIGNATURES

Custodian:

Costin Radulescu, JPL Date

Approval:

PDS Geosciences Node Manager

Raymond Arvidson, Washington University Date

PDS Program Manager

Thomas H. Morgan, PDS Date

Instrument Scientist:

Alberto Behar, JPL Date

OPGS CDE:

Maher Hanna, JPL Date

MSL Deputy Project Scientist

Joy Crisp, JPL Date

Concurrence:

MSL GDS:

Tadas Sesplaukis, JPL, MSL GDS SE Date

Principal Investigator:

Igor Mitrofanov, IKI Date

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
25-May-08	All	First Draft, included K. Harshman, and S. Slaveny comments. Included DAN engineering keywords	Draft
Feb. 02, 2010	All	Incorporated feedback from Karl Harshman, and new information from DAN FDD (Flight Software Team)	Draft
Mar. 8, 2010		Changed structure of label, all DAN HK data present in records was removed from label since they can change from record to record and do not apply to the whole file anymore	0.1
May 5, 2010	Appendix B	Binary data definition to byte level	Draft
June 22, 2010	Appendix A	Updated keywords	Draft
July 20, 2010	Appendix A	Updated keywords	Draft
Dec 10, 2010	Section 5 (PDS label)	Updated Format (.FMT) definitions	1.2 (draft)
Dec 14, 2010	Appendix A	Updated keywords	1.3 (draft)
Feb 25, 2011	All	Incorporated feedback and comments from Susan Slavney (received 1/28/2011)	draft
November 7, 2011	All	Incorporated feedback form the PDS peer review. (Karl Harshman and Costin Radulescu)	Version 1.0
January 31, 2013	Signature page, Appendix-A, Appendix-B	Incorporated updates to the label.	Version 1.0

Table of Contents

ACRONYMS AND ABBREVIATIONS

AMMOS	Advanced Multi-Mission Operations System
ASCII	American Standard Code for Information Interchange
CODMAC	Committee on Data Management and Computation
DAN	Dynamic Albedo of Neutrons
DE	DAN Electronics
DPO	Data Product Object
EDR	Experiment Data Record
FEI	File Exchange Interface
GDS	Ground Data System
IOT	Instrument Operations Team
JPL	Jet Propulsion Laboratory
MIPL	Multi-mission image Processing Laboratory
MPCS	Multi-mission Data Processing and Control System
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
ODS	MSL's Operations Data Store
OPGS	Operations Products Generation Sub-system
PDS	Planetary Data System
RCE	Rover Compute Element
RDR	Reduced Data Record
RTO	Real Time Operations (MSL terminology)
SIS	Software Interface Specification
TBD	To Be Defined/Determined
VICAR	Video image Communications and Retrieval system

1. INTRODUCTION

1.1 Purpose and Scope

The purpose of this Software Interface Specification (SIS) is to provide suppliers and consumers of the Mars Science Lander's (MSL) Dynamic Albedo of Neutrons (DAN) instrument with a detailed description of DAN's Experiment Data Records (EDR). It does not cover any higher-level products, which are collectively known as Reduced Data Records (RDR). They will be covered by a separate SIS, which will be provided by the DAN team.

This SIS includes description of first order DAN science, and the special engineering data products.

The users for whom this SIS is intended for are the scientists who will analyze the data, including those associated with the MSL Project, DAN instrument engineers, and other users in the general planetary science community.

1.2 Contents

This SIS provides a very high level description of how DAN works/operates. It also describes, at high level, how the DAN data product is acquired by the instrument and how it is processed, formatted, labeled, and uniquely identified on the ground.

There are different DAN EDRs with similar general structures but different PDS labels. The EDRs covered by this SIS are based on the instrument's mode of operations, namely standby, passive and active mode.

The document discusses standards used in generating the product and software tools, which may be used to access the information. The data product structure and organization is described in sufficient detail to enable a user to read the product. Finally, an example of the product's PDS label is provided.

1.3 Applicable Documents and Constraints

This Data Product SIS is responsive to the following MSL documents:

1. Mars Science Laboratory Project Archive Generation, Validation and Transfer Plan, Joy Crisp, JPL D-35281, November 13, 2006.
2. Planetary Data System Standards Reference, JPL D-7669 part 2, version 3.8.
3. Planetary Archive Preparation Guide, August 29 2006, http://pds.jpl.nasa.gov.document/apg/index.html.
4. DAN FDD, JPL D-34220, MSL 375-1230
5. Litvak et al. (2008)
6. DAN Science Team and PDS Geosciences Node Interface Control Document (ICD), Version 3.0, July 26, 2010

This SIS is meant to be consistent with the contract negotiated between the MSL Project and the Instrument Principal Investigator (PI) in which reduced data records (EDR/RDR) and documentation (SIS) are explicitly defined as deliverable products.

1.4 Relationships with Other Interfaces

Changes to this DAN SIS document will affect the following products, software, and/or documents.

Table 1: Product and Software Interfaces to this SIS

Name	Type P-product S-software D-document	Owner
DAN EDRs	P	OPGS/MIPL
MSLEdrGen (telemproc)	S	MIPL
MIPL database schema	P	MIPL
Other DAN Programs/Products/Documents	P/S/D	DAN Science Team

2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

2.1 Instrument Overview

The Dynamic Albedo of Neutrons (DAN) is an active/passive neutron spectrometer that measures the abundance and depth distribution of H- and OH-bearing materials (e.g., adsorbed water, hydrated minerals) in a shallow layer (~1 m) of Mars' subsurface along the path of the MSL rover. In active mode, DAN measures the time decay curve (the "dynamic albedo") of the neutron flux from the subsurface induced by its pulsing 14 MeV neutron source (Figure 1 shows an example). A detailed description of the DAN instrument and scientific investigation can be found in Litvak et al. (2008). The experiment is contributed by the Federal Space Agency of Russia.

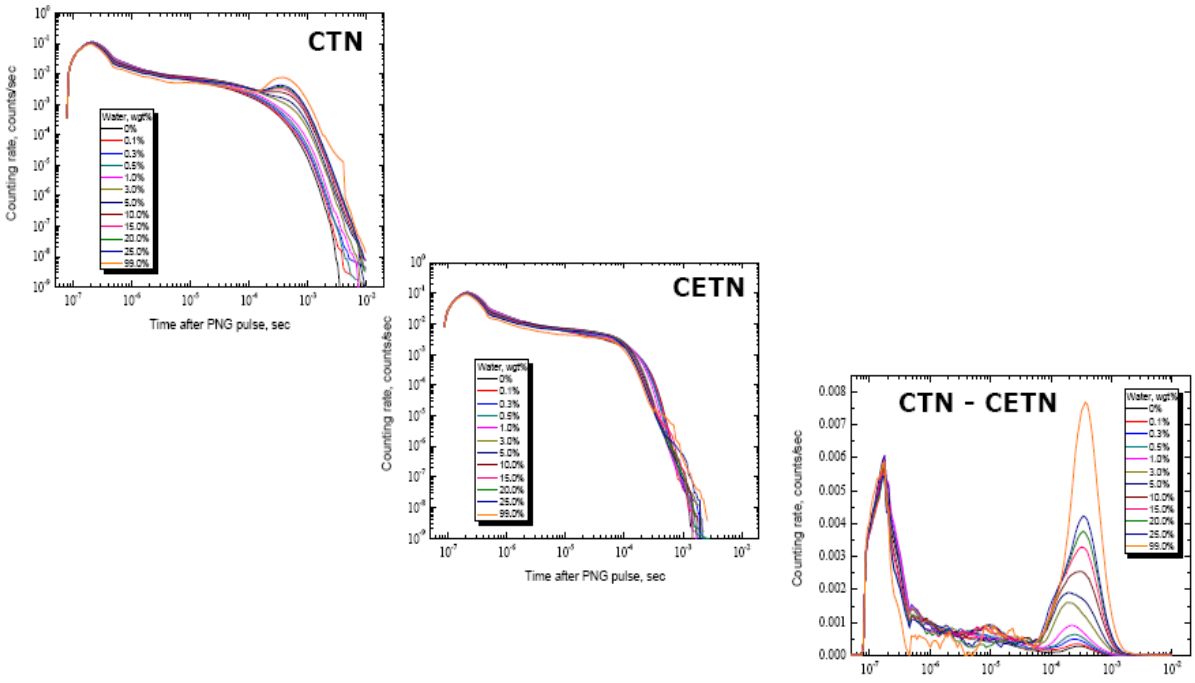


Figure 1: Numerical simulations of the neutron count rate versus time (i.e., die-away curve) for the unshielded (left) and shielded (middle) detectors as a function of water abundance. The right panel shows the difference between the two count rates.

The science objectives of the DAN instrument are as follows: 1) Detect and provide a quantitative estimation of the hydrogen in the subsurface throughout the surface mission; 2) Investigate the upper <0.5 m of the subsurface and determine the possible layering structure of hydrogen-bearing materials in the subsurface; 3) Track the variability of hydrogen content in the upper soil layer (~1 m) during the mission by periodic analysis; and 4) Track the variability of neutron radiation background (neutrons with energy < 100 keV) during the mission by periodic analysis.

The DAN instrument is expected to be used during rover traverses (e.g., during short stops at ~1 m intervals) and while the rover is parked. Short-duration (< 2 min) measurements will provide a rough estimate of the water-equivalent hydrogen distribution with an accuracy of ~1% by weight. Longer-duration (~30 min) measurements are necessary to derive the vertical distribution of water-equivalent hydrogen with an accuracy of 0.1-0.3% by weight.

DAN performs layering structure analyses of the Martian sub-surface, to measure the distribution of H- and OH-bearing materials, with a vertical resolution of < 1 m and horizontal resolutions of 0.5 - 100 m along the path of the rover.

DAN has three different modes of operations: Standby, Passive, and Active.

- STANDBY: Low voltage electronics are on, no science observations.
- PASSIVE: Background observations collected

- ACTIVE: Neutron pulses are produced and science observations collected.

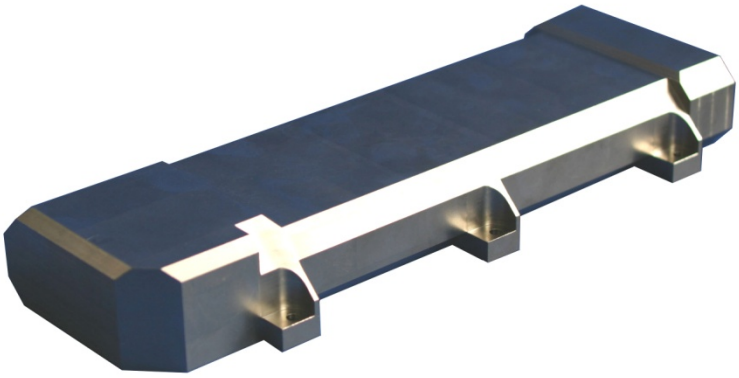


Figure 2: DAN PNG, Pulsing Neutron Generator

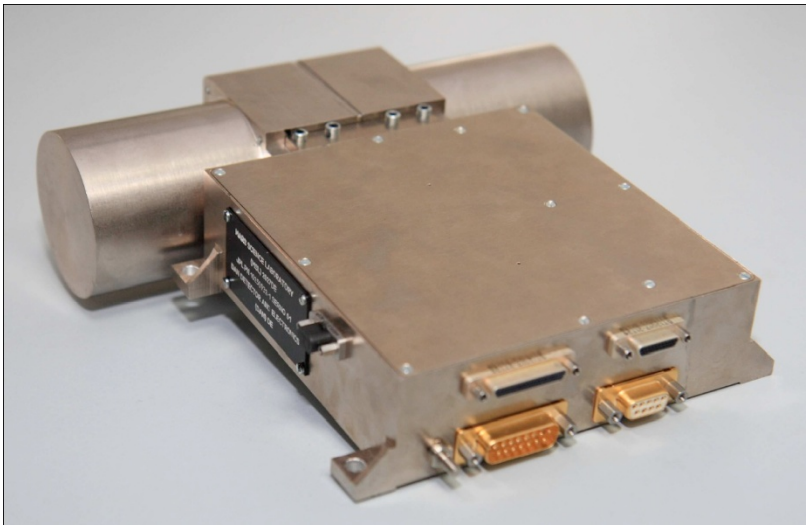


Figure 3: DAN Detector

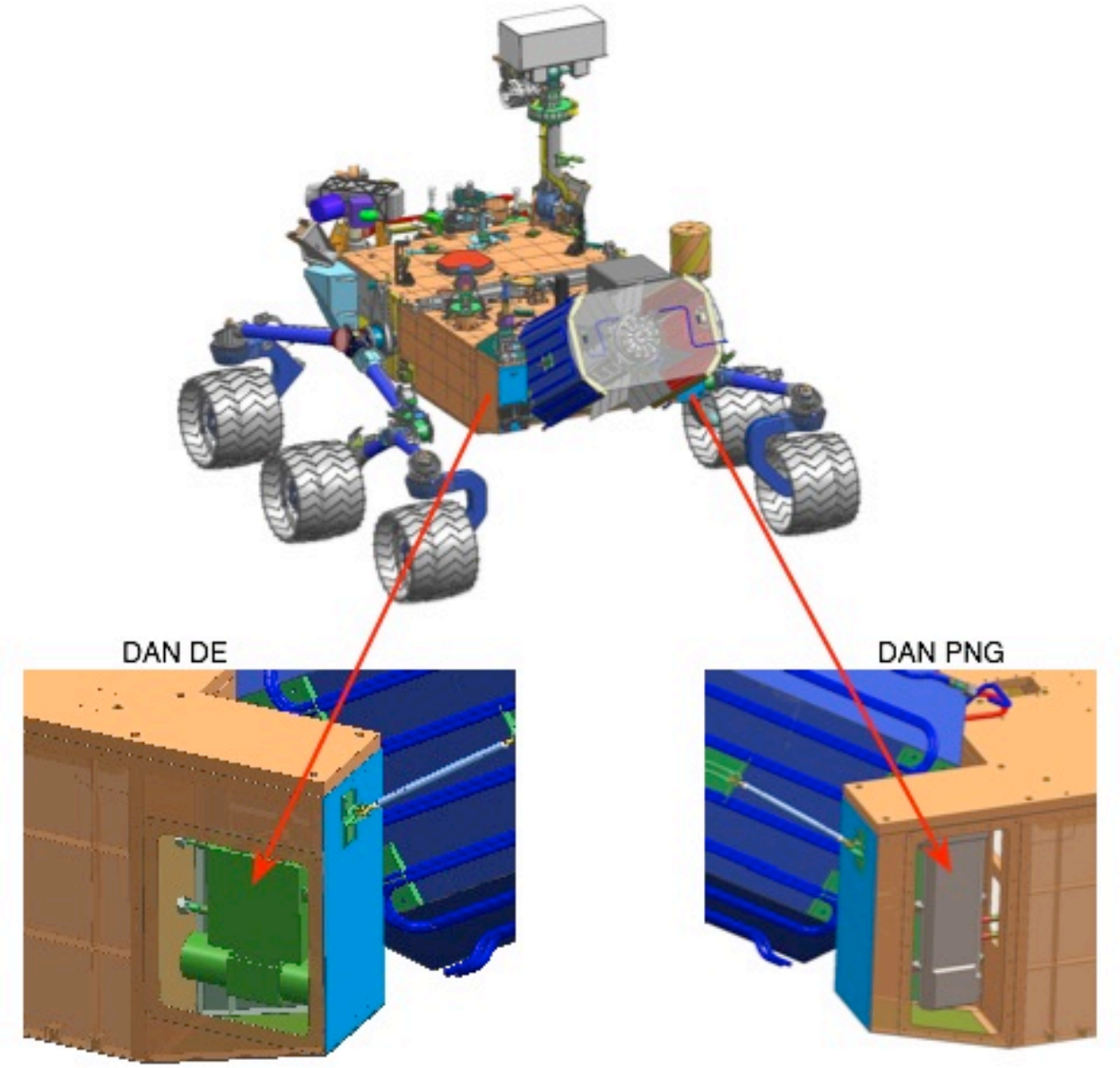


Figure 4: Location of DAN DE and PNG on rover

2.2 Data Product Overview

DAN's Experiment Data Records (EDR), NASA Level 0 products are generated from the reconstructed data products, which are produced by the Multi-Mission Data Processing and Control System (MPCS) on MSL. MPCS is a component of Real Time Operations (RTO), which itself is part of the MSL Ground Data System (GDS).

DAN EDRs consist of clusters of un-calibrated data records, time-ordered, and grouped together. Each EDR will have a unique PDS label file, also known as PDS detached label, which provides all the Meta and Ancillary data for each DAN data product. The pair of files, DAN binary data, and its PDS label, is collectively referred to as an EDR. In other words, an EDR is always made up of two files, named identically with only different file extensions, ".DAT" and ".LBL".

As part of the EDR generation, older or partial EDRs are replaced by newer and more complete versions. Newer versions will always contain the old data, plus any additional new data. EDRs may be used in production of higher-level products (RDRs), which are outside the scope of this document.

Within the Operations Product Generation Sub-system (OPGS), new versions always replace older versions, although outside the OPGS domain, older versions of EDRs may exist.

2.3 Data Processing

2.3.1 Data Processing Level

DAN EDR data products use the "Committee on Data Management and Computation (CODEMAC) data level numbering system. See table below for more details. The scope of this SIS is limited to CODMAC Level 2, NASA level 0.

Table 2: Processing Levels for Science Data Sets

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

NASA	CODMAC	Description
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 DAN EDR data products will be generated by the Multi-mission Image Processing Laboratory (MIPL) at JPL, under the OPGS, using the telemetry processing software called MSLEdrGen. The EDR data products will be raw un-calibrated data reconstructed from telemetry data products and formatted according to this EDR SIS. Meta-data acquired from the telemetry data headers will be used to populate the PDS label. If telemetry packets are missing during the initial downlink from the rover memory, partial data sets will be created. As part of the PDS archive process, typically every 6 month, the entire data set will be reprocessed, and a new or complete version will be created.

With the exception of NAIF provided ancillary data, all Meta data stored in the label is acquired from the telemetry data stream.

2.3.3 Data Flow

The DAN EDR data products generated by MIPL during operations are created collectively from: a) MPCS data products b) SPICE kernels, and c) a meta-data database. They are created on the ODS and then deposited into MIPL's File Exchange Interface (FEI) for electronic distribution to remote sites/users via a secure subscription protocol. After a data validation period, the DAN EDR data products are collected with other science data, and delivered to the Planetary Data System for archiving.

The size of the DAN EDR data file varies depending on the type of data. The DAN data will be reprocessed only if packets in the original downlink are not received. Partial files will be created. The DAN EDR will be reprocessed after all data are retransmitted and received and the original version will be overwritten and placed into FEI for distribution. The remote sites/users will need to obtain and install the FEI client software and set up user access.

2.3.4 Labeling and Identification

There is a file-naming scheme adopted for the MSL image and non-image data products. The scheme applies to the EDR and several RDR data products. The file naming schemes adhere to the Level II 36.3 filename convention approved by PDS in 2009. This is a change from the 27.3 convention that MER and PHX were constrained to using.

The primary attributes of the filename nomenclature are:

a) Uniqueness - It must be unique unto itself without the file system's directory path. This protects against product overwrite as files are copied/moved within the file system and external to the file system, if managed correctly.

b) Metadata - It should be comprised of metadata fields that keep file bookkeeping and sorting intuitive to the human user. Even though autonomous file processing will be managed via databases, there will always be human-in-the-loop that puts a premium on filename intuition. Secondly, the metadata fields should be smartly selected based on their value to ground processing tools, as it is less CPU-intensive to extract information from the filename than from the label.

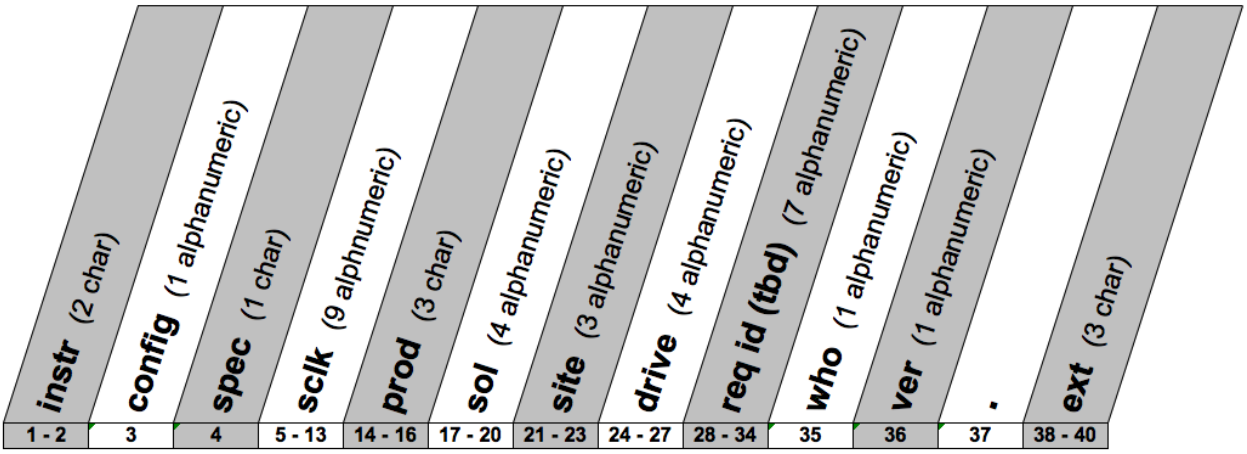
NOTE: The metadata information in the filename also resides in the product label.

The metadata fields have been selected based on MER and PHX lessons learned. In general, the metadata fields are arranged to achieve:

a) Sortability - At the beginning of the filename resides a primary time oriented field such as Spacecraft Clock Start Count (SCLK). This allows for sorting of files on the MSL file system by spacecraft data acquisition time as events occurred on Mars.

b) Readability - An effort is made to alternate Integer fields with ASCII character fields to Optimize differentiation of field boundaries for the human user.

Each DAN EDR has a detached PDS label associated with the DAN data file. The file-naming scheme for the DAN EDR data products is:



where,

instr = (2 alpha character) Instrument ID, denoting the source MSL science or engineering instrument that acquired the data.

Valid values for Instrument ID's are:

"DN" - DAN

config = (1 alphanumeric) Instrument Configuration, an operational attribute of the Instrument that assists in characterizing the data.

Valid values:

Instrument	Configuration	
	Values	Description
DAN ("DN")	It is being set to either "**A**" or "**B**")	A-side configuration, B-side configuration

spec = (1 character) Special Processing flag, applicable to RDRs on a case-by-case basis.

Special Processing	EDR Value	RDR Value
none	"_"	"_"
Special method types A - Z	n/a	"**A"** - "**Z**"

sclk = (9 alphanumeric) Spacecraft Clock Start Count, in units of seconds.

Which specific SCLK is used depends on the instrument but is generally expected to be the time the data was acquired. For the DAN EDRs the SCLK provided in the .emd (Data product metadata file) file will be used.

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

Range 000000000 thru 999999999 - "**000000000**", "**000000001**", **…** "**999999999**"

Range 1000000000 thru 1099999999 - "**A00000000**", "**A00000001**", **…** "**A99999999**"

Range 1100000000 thru 1199999999 - "**B00000000**", **"B00000001**", **…** **"B99999999**"

•
•
•

Range 3500000000 thru 3599999999 - "**Z00000000**", "**Z00000001**", **…** "**Z99999999**"

prod = (3 char) Product Type identifier.

This field has the following rule-of-thumb:

Beginning "**E**" - Type of **E**DR, which is the first order product with no processing applied.

Valid values for Product identifiers are listed below for EDRs:

Product Type Description	Value
Active	"**EAC**"
Passive	"**EPA**"
Standby	"**EST**"

sol = (4 alphanumeric) Sol, or Mars Solar Day. It is converted from the SCLK using LMST (Local Mean Solar Time).

NOTE: If the first character is an underscore ("_") then the remaining 3 characters denote the day of year (DOY). This format will be used during cruise.

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

Range 0000 thru 9999 - "**0000**", "**0001**", **…** "**9999**"
Range 10000 thru 10999 - "**A000**", "**A001**", **…** "**A999**"
Range 11000 thru 11999 - "**B000**", **"B001**", **… "B999**"
•
•
•
Range 35000 thru 35999 - "**Z000**", "**Z001**", **…** "**Z999**"

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

This field has the following rules-of-thumb:

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

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

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

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

This field has the following rules-of-thumb:

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

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

Range 0000 thru 9999 - "**0000**", "**0001**", **…** "**9999**"
Range 10000 thru 10999 - "**A000**", "**A001**", **…** "**A999**"
Range 11000 thru 11999 - "**B000**", **"B001**", **… "B999**"
•
•
•
Range 35000 thru 35999 - "**Z000**", "**Z001**", **…** "**Z999**"

Range 36000 thru 36099 - "**AA00**", "**AA01**", **…** "**AA99**"
Range 36100 thru 36199 - "**AB00**", "**AB01**", **…** "**AB99**"
•
•
•
Range 38500 thru 38599 - "**AZ00**", "**AZ01**", **…** "**AZ99**"
Range 38600 thru 38699 - "**BA00**", "**BA01**", **…** "**BA99**"
Range 38700 thru 38799 - "**BB00**", "**BB01**", **…** "**BB99**"
•
•
•
Range 41100 thru 41199 - "**BZ00**", "**BZ01**", **…** "**BZ99**"
Range 41200 thru 41299 - "**CA00**", "**CA01**", **…** "**CA99**"
•
•
•
Range 65400 thru 65499 - "**LI00**", "**LI01**", **…** "**LI99**"
Range 65500 thru 65535 - "**LJ00**", "**LJ01**", **…** "**LJ35**"

req id = (7 alphanumeric) - Request ID

In the case of DAN this files is currently not being used, and therefore it will always have a value of all underscores ("_____"). It is maintained as a placeholder, and to maintain consistency with MSL file-naming nomenclature.

Valid values:

"_____" – N/A.

who = (1 alpha character) Product Producer ID identifies the institution that generated the product.

This field has the following rules-of-thumb:

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

Valid values:

"M" – MIPL (at JPL)

This field is consistent with the rest of the MSL instruments file-naming convention scheme.

ver = (1 alphanumeric) Version identifier.

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

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

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

version of that product.

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

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

Valid values for nominal operations data products:

"**DAT**" - File contains data, most likely in a binary format
"**LBL**" - Detached label in PDS format.

Example #1: DNA_351797691EPA_0550000000_____M1.DAT

Where,

instr	=	"**DN**"	=	DAN
config	=	"**A**"	=	Instrument configuration
spec	=	"**_**"	=	Unused
sclk	=	"**_**"	=	Spacecraft Clock Start Count of 351797691 sec.
prod	=	"**EPA**"	=	"E"-EDR, "PA"-Passive
sol	=	"**_055**"	=	Day-of-Year 55 (Cruise)
site	=	"**000**"	=	Site 0
drive	=	"**0000**"	=	Position (Drive) 0
requestID	=	"**_____**"	=	Unused
who	=	"**M**"	=	Produced by **M**IPL at JPL
ver	=	"**1**"	=	Version 1
ext	=	"**DAT**"	=	Indicates this file contains data, and not label info (LBL).

2.4 Standards Used in Generating Data Products

2.4.5 PDS Standards

The DAN data product complies with PDS standards for file formats and labels, as specified in the PDS Standards Reference version 3.8. It also adheres to the Level II, 36.3-filename conventions for PDS standards compliance.

2.4.6 Time Standards

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

Time Format	*Definition*
SCET	Spacecraft event time. This is the time when an event occurred on-board the spacecraft, in UTC. It is usually derived from SCLK.

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

2.4.7 Coordinate Systems

The following coordinate systems are used within the project to refer to the position of the Lander and it's instruments.

Coordinate System	***Origin***	***Orientation***
Local Level	Same as payload frame, and it moves with the Lander	+X North +Z down along gravity vector +Y East
Payload Frame	At the shoulder of the Robotic Arm. Attached and moves with the Lander	+X along Lander –X (point out into the work space) +Z down along Lander (vertical

		axis) +Y along Lander -Y
Site Frame	Same as payload frame when first defined and never moves relative to Mars. Possible to define multiple site frames in case the Lander moves/slips.	Same as local level
Rover Frame	Attached to Rover	Aligned with Rover

2.4.8 Data Storage Conventions

DAN EDR data files (.DAT) products will be written as little-endian binary files. For every DAN EDR, there will be a detached PDS label file. The detached PDS labels for DAN EDRs are stored as ASCII text, with each keyword definition terminated by ASCII carriage-return and line-feed characters. The EDR products are described/defined as PDS table objects. The PDS label keywords are described in Appendix-A

All DAN EDRs will contain fixed length records, although the size of the records in each file could differ between EDRs. Label keywords will provide necessary information to determine the size and organization of the binary records.

2.5 Data Validation

The DAN EDRs, as with all other MSL EDRs, are subject to PDS peer review.

Validation of DAN EDR products during production will be performed according to specifications in the MSL Archive Plan and the DAN-PDS Interface Control Document. The DAN Team will validate the science content of the data products, and the PDS Geosciences Node will validate the products for compliance with PDS standards and for conformance with the design specified in this SIS.

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

Automated validation will be performed as a part of the archiving process, and will be done simultaneously with the archive volume validation. Validations performed, will include such things as verification that the checksum in the label matches a calculated checksum for the data product (i.e., that the data product included in the archive is identical to that produced by the real-time process), a validation of the PDS syntax of the label, a check of the label values against the database and against the index tables included on the archive volume, and checks for internal consistency of the label items. The latter include such things as verifying that the product creation date is later than the earth received time. 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). A human will view these products. The DAN Team will validate the science content of the data products, and the corresponding PDS Geosciences Node will validate the products for compliance with PDS standards and for conformance with the design specified in this SIS.

3. DETAILED DATA PRODUCT SPECIFICATIONS

When powered, the DAN instrument is in a one of three states: STANDBY, PASSIVE, and ACTIVE. Onboard, the DAN instruments can generate different data packets depending on the state of the instrument.

The instrument sends its data out, in the form of packets to the Rover Compute Element (RCE), which in turn builds them into discreet data products. The different DAN data products are listed, in reference #5 (DAN FDD MSL-375-1230).

To facilitate ground processing, each of the above data products will be saved as a different EDR. Each are defined in the following sections. The DAN EDR file naming convention will uniquely identify each data product. See Section 2.3.4 for detail description of the DAN EDR file naming convention.

The structure of the DAN EDR consists of a detached ASCII PDS label and a binary data file as shown in Figure 3.1.

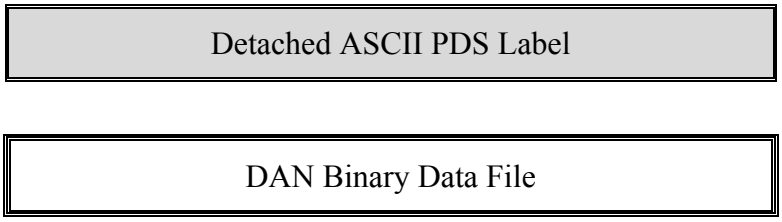


Figure 3.1: The DAN EDR consists of two files.

3.1 Data Format Descriptions

Section 5.1.1 below describes the binary data file, as PDS column definitions, to the byte level.

3.2 Label and Header Descriptions

3.2.1 PDS Label

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

^Object = location

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

Each PDS keywords defined for the DAN label will always be included in the PDS label. If a keyword does not have a value, a value of N/A will be given as the keyword value.

Per PDS rules, “N/A” is used when a keyword exists, but it does not apply to a particular data, and “UNKNOWN” is used when a value of an applicable keyword cannot be determined at the time the PDS label was generated.

3.2.2 VICAR Label

Does not apply to DAN products.

3.2.3 PDS Table Object

The TABLE object is a uniform collection of rows containing binary values stored in columns. The INTERCHANGE_FORMAT keyword is used to distinguish between TABLEs containing only ASCII columns and those containing binary data. The rows and columns of the TABLE object provide a natural correspondence to the records and fields often defined in interface specifications for existing data products. Each field is defined as a fixed-width COLUMN object; the value of the COLUMNS keyword is the total number of COLUMN objects defined in the label. All TABLE objects must have fixed-width records.

4. APPLICABLE SOFTWARE

4.1 Utility Programs

None

4.2 Applicable PDS Software Tools

PDS-labeled images and tables can be viewed with the program NASAView, developed by the PDS engineering node, and are available for a variety of computer platforms. They can be obtained directly from the PDS, web site http://pdsproto.jpl.nasa.gov/Distribution/license.html.

5. EXAMPLE OF A DAN PDS LABEL

5.1 Passive Label

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 208
FILE_RECORDS                    = 180

/* POINTERS TO DATA OBJECTS */

^SCIENCE_TABLE                  =
("DNB_417353685EPA02240000000_______M1.DAT",1)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-DAN-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS DYNAMIC ALBEDO OF NEUTRONS 2
EDR V1.0"
COMMAND_SEQUENCE_NUMBER         = 0
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = DAN
INSTRUMENT_NAME                 = "DYNAMIC ALBEDO OF NEUTRONS"
INSTRUMENT_TYPE                 = SPECTROMETER
LOCAL_MEAN_SOLAR_TIME           = "Sol-00224M05:52:37:031"
LOCAL_TRUE_SOLAR_TIME           = "05:21:27"
MISSION_NAME                    = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME              = TEST
OBSERVATION_ID                  = UNK
PLANET_DAY_NUMBER               = 224
PRODUCER_INSTITUTION_NAME       = "MULTIMISSION IMAGE PROCESSING
LABORATORY
                                  , JET PROPULSION LAB"
PRODUCT_CREATION_TIME           = 2012-02-03T19:15:31.000
PRODUCT_ID                      = "DNB_417353685EPA02240000000_______M1"
PRODUCT_VERSION_ID              = "V1.0 D-22849"
PRODUCT_TYPE                    = DAN_PASSIVE
RELEASE_ID                      = "0001"
MSL:REQUEST_ID                  = 0
ROVER_MOTION_COUNTER            = (0,0,0,0,0,0,0,0)
ROVER_MOTION_COUNTER_NAME       = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                  DRILL, RSM, HGA)
SEQUENCE_ID                     = "1036288"
```

```
SEQUENCE_VERSION_ID              = "0"
MSL:ACTIVE_FLIGHT_STRING_ID      = B
SOLAR_LONGITUDE                  = -72.1967
SPACECRAFT_CLOCK_CNT_PARTITION   = 1
SPACECRAFT_CLOCK_START_COUNT     = "0417353685.048"
SPACECRAFT_CLOCK_STOP_COUNT      = ""
START_TIME                       = ""
STOP_TIME                        = ""
TARGET_NAME                      = MARS
TARGET_TYPE                      = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID           = 173
APPLICATION_PROCESS_NAME         = "DAN_PASSIVE"
MSL:VIRTUAL_CHANNEL_ID           = 32
MSL:COMMUNICATION_SESSION_ID     = "0"
DOWNLOAD_PRIORITY                = 31
EARTH_RECEIVED_START_TIME        = 2011-03-23T02:26:18.324
MSL:EXPECTED_TRANSMISSION_PATH   = "3851"
MSL:AUTO_DELETE_FLAG             = "0"
MSL:TRANSMISSION_PATH            = 65535
FLIGHT_SOFTWARE_VERSION_ID       = "97208714"
MSL:FLIGHT_SOFTWARE_MODE         = "8"
MSL:PRODUCT_COMPLETION_STATUS    = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                  = "0"
MSL:SEQUENCE_EXECUTION_COUNT     = 0
RECEIVED_PACKETS                 = 4
SPICE_FILE_NAME                  = "chronos.msl"
TELEMETRY_PROVIDER_ID            = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM    = 20291
MSL:TELEMETRY_SOURCE_SCLK_START  = "417353685.48099"
MSL:TELEMETRY_SOURCE_SIZE        = 37440
MSL:TELEMETRY_SOURCE_START_TIME  = 2013-082T23:33:39.550
TELEMETRY_SOURCE_NAME            = "DanPassive_0417353685_73393-1.dat"
MSL:TELEMETRY_SOURCE_TYPE        = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME   = mslatlompcs4
EXPECTED_PACKETS                 = 4

/* SOURCE DATA ELEMENTS */

SOFTWARE_NAME                    = “msledrgen”
SOFTWARE_VERSION_ID              = “1.0”

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

GROUP                            = ROVER_COORDINATE_SYSTEM
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (0,0,0,0,0,0,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (0,0,0,0)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
```

```
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX    = 0
END_GROUP                        = ROVER_COORDINATE_SYSTEM

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */

GROUP                            = RSM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = RSM
 ARTICULATION_DEVICE_NAME        = REMOTE_SENSING_MAST
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = (AZIMUTH, ELEVATION)
END_GROUP                        = RSM_ARTICULATION_STATE

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

GROUP                            = ARM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>, 0<rad>, 0<rad>,
0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                    "JOINT 2 SHOULDER ELEVATION",
                                    "JOINT 3 ELBOW-ENCODER",
                                    "JOINT 4 WRIST-ENCODER",
                                    "JOINT 5 TURRET-ENCODER")
END_GROUP                        = ARM_ARTICULATION_STATE



/* DATA OBJECT DEFINITION DESCRIPTION */

OBJECT                           = SCIENCE_TABLE
   COLUMNS                       = 76
   INTERCHANGE_FORMAT            = BINARY
   ROW_BYTES                     = 208
   ROWS                          = 180
   DESCRIPTION                   = "
    This table contains neutron spectra and associated instrument
parameters
    as observed by the Mars Science Laboratory (MSL) Dynamic Albedo of
    Neutron Detector (DAN).

    Detailed descriptions for the parameters defined below are
    contained in the DAN EDR SIS document.

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

5.1.1 Passive FORMAT FILE ("DAN_EDR_PASSIV.FMT")

```
OBJECT = COLUMN
   NAME = SCLK
   COLUMN_NUMBER = 1
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 1
   DESCRIPTION = "The Rover Compute Element (RCR) clock in seconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = AUTOBYTE_CNT
   COLUMN_NUMBER = 2
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 5
   DESCRIPTION = "The number of bytes in the data packet. This includes
FRAME_HDR
   through DAN_CHECKSUM"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FRAME_HDR
   COLUMN_NUMBER = 3
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 9
   DESCRIPTION = "The data frame header. This is broken down into the
following bits:
   bits 1 - 8 is opcode which specifies an instrument activity,
   bits 9 - 10 specifies error control algorithm, DAN uses the checksum
error control,
   bit 11 indicates the presence (1) or absense (0) of both DATA_LEN and
DATA.
   bits 12 - 32 are a status flag defined as follows:
   bit 12 is on if instrument in standby mode,
   bit 13 is on if instrument in passive mode
   bit 14 is on if instrument in active mode
   bit 15 is on indicates PNG warning, an anomily with PNG RCE to turn off
PNG
   bit 16 is on if instrument has taken interesting data
   bits 17 - 32 are unused."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_LEN
   COLUMN_NUMBER = 4
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 13
   DESCRIPTION = "The length of the data packet from the instrument
   in bytes. This includes column SYNCRO through HV_VALUES."
END_OBJECT = COLUMN
```

```
OBJECT = COLUMN
   NAME = SYNCRO
   COLUMN_NUMBER = 5
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 17
   DESCRIPTION = "The synchronization bytes at the beginning of each
record."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_FRAME_NUMBER
   COLUMN_NUMBER = 6
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 21
   DESCRIPTION = "Incrementing counter for data packets."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_TIME
   COLUMN_NUMBER = 7
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 23
   DESCRIPTION = "DAN instrument time in milliseconds since last
                  power up."
END_OBJECT = COLUMN

OBJECT = CONTAINER
   NAME = CMDS_ARRAY
   BYTES = 8
   START_BYTE = 27
   REPETITIONS = 8
   DESCRIPTION = "The last 8 commands sent to the instrument."

   OBJECT = COLUMN
      COLUMN_NUMBER = 8
      NAME = COMMAND_TIME
      BYTES = 4
      START_BYTE = 1
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "DAN instrument time in milliseconds when command
                     was executed."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 9
      NAME = OPCODE
      BYTES = 1
      START_BYTE = 5
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Command identifier."
   END_OBJECT = COLUMN
```

```
   OBJECT = COLUMN
      COLUMN_NUMBER = 10
      NAME = PARAMS
      BYTES = 1
      START_BYTE = 6
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Bit field:
         - bit 7: data present bit, when '0' argument1 and argument2
                  datafields are meaningless
         - bit 6 and 5 are meaningless
         - bits 4 through 0: condition code, shows status of the command
                  execution according to electrical specs."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 11
      NAME = ARG1
      BYTES = 1
      START_BYTE = 7
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "First argument to the command."
    END_OBJECT = COLUMN

    OBJECT = COLUMN
      COLUMN_NUMBER = 12
      NAME = ARG2
      BYTES = 1
      START_BYTE = 8
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Second argument to the command."
    END_OBJECT = COLUMN

END_OBJECT = CONTAINER

OBJECT = COLUMN
   NAME = FRAME_TYPE
   COLUMN_NUMBER = 48
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 91
   DESCRIPTION = "Current instrument mode, 0 - standby,
                  1 - passive, 2 - active."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP1
   COLUMN_NUMBER = 49
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 92
   DESCRIPTION = "PNG temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP2
```

```
   COLUMN_NUMBER = 50
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 93
   DESCRIPTION = "High voltage board temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP3
   COLUMN_NUMBER = 51
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 94
   DESCRIPTION = "FPGA temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP4
   COLUMN_NUMBER = 52
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 95
   DESCRIPTION = "Analog board #1 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP5
   COLUMN_NUMBER = 53
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 96
   DESCRIPTION = "Analog board #2 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP6
   COLUMN_NUMBER = 54
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 97
   DESCRIPTION = "Case temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_CHARGE_TIME
   COLUMN_NUMBER = 55
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 98
   DESCRIPTION = "PNG charge time in milliseconds"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_PULSES
   COLUMN_NUMBER = 56
```

```
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 99
   DESCRIPTION = "Number of PNG pulses since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_CMD_RECV
   COLUMN_NUMBER = 57
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 101
   DESCRIPTION = "Number of commands received since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_CMD_RJCT
   COLUMN_NUMBER = 58
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 103
   DESCRIPTION = "Number of commands rejected since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_ALARM
   COLUMN_NUMBER = 59
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 105
   DESCRIPTION = "Indication of PNG Alarm, bifield:
                  Bit 7: Alarm by PNG HV out of range
                  Bit 6: PNG temperature out of range
                  Bit 5: Detector 2 anomaly countrate
                  Bit 4: Detector 1 anomaly countrate
                  Bit 0: PNG alarm detected."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = RESET_TYPE
   COLUMN_NUMBER = 60
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 106
   DESCRIPTION = "Type of reset:
                   0 - Power on reset
                   1 - RS-422 cmd reset
                   2 - Discrete line A reset
                   3 - Discrete line B reset."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = HV_VALUES
   COLUMN_NUMBER = 61
   BYTES = 16
```

```
   ITEMS = 16
   ITEM_BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 107
   DESCRIPTION = "History of the PNG high voltage, last 16 values.
   Real value in Volts calculated by 1 / (17 * value / 1600000)"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_NORM_PULSES
   COLUMN_NUMBER = 62
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 123
   DESCRIPTION = "Number of normal pulses accumulated in CTN and CETN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_SPONTANEOUS_PULSES
   COLUMN_NUMBER = 63
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 125
   DESCRIPTION = "Number of spontaneous pulses not accumulated."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_MISSED_PULSES
   COLUMN_NUMBER = 64
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 126
   DESCRIPTION = "Number of pulses that did not take place."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = ACCUM_TIME
   COLUMN_NUMBER = 65
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 127
   DESCRIPTION = "Accumulation time for measurement in 1/10sec."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_FREQUENCY
   COLUMN_NUMBER = 66
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 129
   DESCRIPTION = "Pulsing Neutron Generator frequency."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = LEVELS
```

```
   COLUMN_NUMBER = 67
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 130
   DESCRIPTION = "Bits 7:6 = High Voltage 1.
                       0 = off, 2 = high, 3 = low.
                 Bits 5:4 = High Voltage 2.
                       0 = off, 2 = high, 3 = low.
                 Bits 3:2 = Discriminator level 1.
                       0 = low, 1 = high.
                 Bits 1:0 = Discriminator level 2.
                       0 = low, 1 = high."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = BIN_TIME
   COLUMN_NUMBER = 68
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 131
   DESCRIPTION = "basic binning time for pulses, divide by 2 to
      get microseconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = BIN_SCALING
   COLUMN_NUMBER = 69
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 132
   DESCRIPTION = "Multiplier of how much larger next bin will be."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PULSE_TIME
   COLUMN_NUMBER = 70
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 133
   DESCRIPTION = "Pulsing time in 1/10sec."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = MAX_PULSES
   COLUMN_NUMBER = 71
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 135
   DESCRIPTION = "Maximum number of pulses."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_PROTECTION
   COLUMN_NUMBER = 72
   BYTES = 2
```

```
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 137
   DESCRIPTION = "PNG protection options. Least Significant Bit is bit 0.
                  When bits 0-3 are on, bits 8-15 are ignored
                  bits 11:08 = protection threshold for detector #1.
                  bits 15:12 = protection threshold for detector #2.
                  value 0 = highest, 15 = lowest 15 is lowest protection.
                  bit 3 = '1' to disable warning from PNG HV measurements
                  bit 2 = '1' to disable warning from PNG temperature
sensors
                  bit 1 = '1' to disable warning from detector #2
                  bit 0 = '1' to disable warning from detector #1"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = CTN_SPECTRUM
   COLUMN_NUMBER = 73
   BYTES = 32
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 139
   ITEMS = 16
   ITEM_BYTES = 2
   DESCRIPTION = "A single 16 channel CTN spectrum over time in
ACCUM_TIME.
   Epithermal neutrons from 0.4 eV up to energy of about 100 keV."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = CETN_SPECTRUM
   COLUMN_NUMBER = 74
   BYTES = 32
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 171
   ITEMS = 16
   ITEM_BYTES = 2
   DESCRIPTION = "A single 16 channel CETN spectrum over time in
ACCUM_TIME.
   Epithermal and thermal neutrons up to energy of about 100 keV."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_CHECKSUM
   COLUMN_NUMBER = 75
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 123
   DESCRIPTION = "Checksum at end of instrument data, lowest two bytes
   of the sum of all bytes in data."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FLETCH_CHECKSUM
   COLUMN_NUMBER = 76
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
   START_BYTE = 205
   DESCRIPTION = "Fletcher checksum added by rover processor."
END_OBJECT = COLUMN
```

5.2 Active Label

Same keywords as in the Passive Label (Section 5.1), except the column definitions (.FMT file) are as described in Section 5.2.1 below.

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 4240
FILE_RECORDS                    = 30

/* POINTERS TO DATA OBJECTS */

^SCIENCE_TABLE                  =
("DNB_417337557EAC02240000000_______M1.DAT",1)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-DAN-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS DYNAMIC ALBEDO OF NEUTRONS 2
EDR V1.0"
COMMAND_SEQUENCE_NUMBER         = 5
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = DAN
INSTRUMENT_NAME                 = "DYNAMIC ALBEDO OF NEUTRONS"
INSTRUMENT_TYPE                 = SPECTROMETER
LOCAL_MEAN_SOLAR_TIME           = "Sol-00224M01:31:00:068"
LOCAL_TRUE_SOLAR_TIME           = "00:59:57"
MISSION_NAME                    = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME              = TEST
OBSERVATION_ID                  = UNK
PLANET_DAY_NUMBER               = 224
PRODUCER_INSTITUTION_NAME       = "MULTIMISSION IMAGE PROCESSING
LABORATORY
                                  , JET PROPULSION LAB"
PRODUCT_CREATION_TIME           = 2012-02-03T19:14:56.000
PRODUCT_ID                      = "DNB_417337557EAC02240000000_______M1"
PRODUCT_VERSION_ID              = "V1.0 D-22849"
PRODUCT_TYPE                    = DAN_ACTIVE
RELEASE_ID                      = "0001"
MSL:REQUEST_ID                  = 0
ROVER_MOTION_COUNTER            = (0,0,0,0,0,0,0,0)
ROVER_MOTION_COUNTER_NAME       = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                  DRILL, RSM, HGA)
SEQUENCE_ID                     = "213806"
SEQUENCE_VERSION_ID             = "2"
```

```
MSL:ACTIVE_FLIGHT_STRING_ID      = B
SOLAR_LONGITUDE                  = -72.3112
SPACECRAFT_CLOCK_CNT_PARTITION   = 1
SPACECRAFT_CLOCK_START_COUNT     = "0417337557.002"
SPACECRAFT_CLOCK_STOP_COUNT      = ""
START_TIME                       = ""
STOP_TIME                        = ""
TARGET_NAME                      = MARS
TARGET_TYPE                      = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID           = 170
APPLICATION_PROCESS_NAME         = "DAN_ACTIVE"
MSL:VIRTUAL_CHANNEL_ID           = 32
MSL:COMMUNICATION_SESSION_ID     = "0"
DOWNLOAD_PRIORITY                = 31
EARTH_RECEIVED_START_TIME        = 2011-03-22T21:07:21.208
MSL:EXPECTED_TRANSMISSION_PATH   = "3851"
MSL:AUTO_DELETE_FLAG             = "0"
MSL:TRANSMISSION_PATH            = 65535
FLIGHT_SOFTWARE_VERSION_ID       = "97208714"
MSL:FLIGHT_SOFTWARE_MODE         = "8"
MSL:PRODUCT_COMPLETION_STATUS    = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                  = "0"
MSL:SEQUENCE_EXECUTION_COUNT     = 1
RECEIVED_PACKETS                 = 15
SPICE_FILE_NAME                  = "chronos.msl"
TELEMETRY_PROVIDER_ID            = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM    = 64005
MSL:TELEMETRY_SOURCE_SCLK_START  = "417337557.2941"
MSL:TELEMETRY_SOURCE_SIZE        = 127200
MSL:TELEMETRY_SOURCE_START_TIME  = 2013-082T19:04:50.861
TELEMETRY_SOURCE_NAME            = "DanActive_0417337557_04488-1.dat"
MSL:TELEMETRY_SOURCE_TYPE        = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME   = mslatlompcs4
EXPECTED_PACKETS                 = 15

/* SOURCE DATA ELEMENTS */

SOFTWARE_NAME                    = “msledrgen”
SOFTWARE_VERSION_ID              = “1.0”

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

GROUP                            = ROVER_COORDINATE_SYSTEM
 COORDINATE_SYSTEM_NAME          = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX         = (0,0,0,0,0,0,0,0)
 COORDINATE_SYSTEM_INDEX_NAME    = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                   DRILL, RSM, HGA)
 ORIGIN_OFFSET_VECTOR            = (0,0,0)
 ORIGIN_ROTATION_QUATERNION      = (0,0,0,0)
 POSITIVE_AZIMUTH_DIRECTION      = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION    = UP
 REFERENCE_COORD_SYSTEM_NAME     = SITE_FRAME
```

```
 REFERENCE_COORD_SYSTEM_INDEX    = 0
END_GROUP                        = ROVER_COORDINATE_SYSTEM

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */

GROUP                            = RSM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = RSM
 ARTICULATION_DEVICE_NAME        = REMOTE_SENSING_MAST
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = (AZIMUTH, ELEVATION)
END_GROUP                        = RSM_ARTICULATION_STATE

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

GROUP                            = ARM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>, 0<rad>, 0<rad>,
0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                    "JOINT 2 SHOULDER ELEVATION",
                                    "JOINT 3 ELBOW-ENCODER",
                                    "JOINT 4 WRIST-ENCODER",
                                    "JOINT 5 TURRET-ENCODER")
END_GROUP                        = ARM_ARTICULATION_STATE



/* DATA OBJECT DEFINITION DESCRIPTION */

OBJECT                           = SCIENCE_TABLE
   COLUMNS                       = 76
   INTERCHANGE_FORMAT            = BINARY
   ROW_BYTES                     = 4240
   ROWS                          = 30
   DESCRIPTION                   = "
    This table contains neutron spectra and associated instrument
parameters
    as observed by the Mars Science Laboratory (MSL) Dynamic Albedo of
    Neutron Detector (DAN).

    Detailed descriptions for the parameters defined below are
    contained in the DAN EDR SIS document.

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

5.2.1 Active FORMAT FILE ("DAN_EDR_ACTIVV.FMT")

```
OBJECT = COLUMN
```

```
   NAME = SCLK
   COLUMN_NUMBER = 1
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 1
   DESCRIPTION = "The Rover Compute Element (RCR) clock in seconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = AUTOBYTE_CNT
   COLUMN_NUMBER = 2
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 5
   DESCRIPTION = "The number of bytes in the data packet. This includes
FRAME_HDR
   through DAN_CHECKSUM"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FRAME_HDR
   COLUMN_NUMBER = 3
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 9
   DESCRIPTION = "The data frame header. This is broken down into the
following bits:
   bits 1 - 8 is opcode which specifies an instrument activity,
   bits 9 - 10 specifies error control algorithm, DAN uses the checksum
error control,
   bit 11 indicates the presence (1) or absense (0) of both DATA_LEN and
DATA.
   bits 12 - 32 are a status flag defined as follows:
   bit 12 is on if instrument in standby mode,
   bit 13 is on if instrument in passive mode
   bit 14 is on if instrument in active mode
   bit 15 is on indicates PNG warning, an anomily with PNG RCE to turn off
PNG
   bit 16 is on if instrument has taken interesting data
   bits 17 - 32 are unused."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_LEN
   COLUMN_NUMBER = 4
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 13
   DESCRIPTION = "The length of the data packet from the instrument
   in bytes. This includes column SYNCRO through HV_VALUES."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = SYNCRO
   COLUMN_NUMBER = 5
```

```
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 17
   DESCRIPTION = "The synchronization bytes at the beginning of each
record."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_FRAME_NUMBER
   COLUMN_NUMBER = 6
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 21
   DESCRIPTION = "Incrementing counter for data packets."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_TIME
   COLUMN_NUMBER = 7
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 23
   DESCRIPTION = "DAN instrument time in milliseconds since last
                  power up."
END_OBJECT = COLUMN

OBJECT = CONTAINER
   NAME = CMDS_ARRAY
   BYTES = 8
   START_BYTE = 27
   REPETITIONS = 8
   DESCRIPTION = "The last 8 commands sent to the instrument."

   OBJECT = COLUMN
      COLUMN_NUMBER = 8
      NAME = COMMAND_TIME
      BYTES = 4
      START_BYTE = 1
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "DAN instrument time in milliseconds when command
                     was executed."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 9
      NAME = OPCODE
      BYTES = 1
      START_BYTE = 5
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Command identifier."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 10
      NAME = PARAMS
```

```
      BYTES = 1
      START_BYTE = 6
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Bit field:
         - bit 7: data present bit, when '0' argument1 and argument2
                  datafields are meaningless
         - bit 6 and 5 are meaningless
         - bits 4 through 0: condition code, shows status of the command
                  execution according to electrical specs."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 11
      NAME = ARG1
      BYTES = 1
      START_BYTE = 7
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "First argument to the command."
    END_OBJECT = COLUMN

    OBJECT = COLUMN
      COLUMN_NUMBER = 12
      NAME = ARG2
      BYTES = 1
      START_BYTE = 8
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Second argument to the command."
    END_OBJECT = COLUMN

END_OBJECT = CONTAINER

OBJECT = COLUMN
   NAME = FRAME_TYPE
   COLUMN_NUMBER = 48
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 91
   DESCRIPTION = "Current instrument mode, 0 - standby,
                 1 - passive, 2 - active."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP1
   COLUMN_NUMBER = 49
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 92
   DESCRIPTION = "PNG temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP2
   COLUMN_NUMBER = 50
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
   START_BYTE = 93
   DESCRIPTION = "High voltage board temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP3
   COLUMN_NUMBER = 51
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 94
   DESCRIPTION = "FPGA temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP4
   COLUMN_NUMBER = 52
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 95
   DESCRIPTION = "Analog board #1 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP5
   COLUMN_NUMBER = 53
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 96
   DESCRIPTION = "Analog board #2 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP6
   COLUMN_NUMBER = 54
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 97
   DESCRIPTION = "Case temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_CHARGE_TIME
   COLUMN_NUMBER = 55
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 98
   DESCRIPTION = "PNG charge time in milliseconds"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_PULSES
   COLUMN_NUMBER = 56
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 99
```

```
   DESCRIPTION = "Number of PNG pulses since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_CMD_RECV
   COLUMN_NUMBER = 57
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 101
   DESCRIPTION = "Number of commands received since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_CMD_RJCT
   COLUMN_NUMBER = 58
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 103
   DESCRIPTION = "Number of commands rejected since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_ALARM
   COLUMN_NUMBER = 59
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 105
   DESCRIPTION = "Indication of PNG Alarm, bifield:
                  Bit 7: Alarm by PNG HV out of range
                  Bit 6: PNG temperature out of range
                  Bit 5: Detector 2 anomaly countrate
                  Bit 4: Detector 1 anomaly countrate
                  Bit 0: PNG alarm detected."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = RESET_TYPE
   COLUMN_NUMBER = 60
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 106
   DESCRIPTION = "Type of reset:
                   0 - Power on reset
                   1 - RS-422 cmd reset
                   2 - Discrete line A reset
                   3 - Discrete line B reset."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = HV_VALUES
   COLUMN_NUMBER = 61
   BYTES = 16
   ITEMS = 16
   ITEM_BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
   START_BYTE = 107
   DESCRIPTION = "History of the PNG high voltage, last 16 values.
   Real value in Volts calculated by 1 / (17 * value / 1600000)"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_NORM_PULSES
   COLUMN_NUMBER = 62
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 123
   DESCRIPTION = "Number of normal pulses accumulated in CTN and CETN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_SPONTANEOUS_PULSES
   COLUMN_NUMBER = 63
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 125
   DESCRIPTION = "Number of spontaneous pulses not accumulated."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_MISSED_PULSES
   COLUMN_NUMBER = 64
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 126
   DESCRIPTION = "Number of pulses that did not take place."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = ACCUM_TIME
   COLUMN_NUMBER = 65
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 127
   DESCRIPTION = "Accumulation time for measurement in 1/10sec."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_FREQUENCY
   COLUMN_NUMBER = 66
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 129
   DESCRIPTION = "Pulsing Neutron Generator frequency."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = LEVELS
   COLUMN_NUMBER = 67
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
   START_BYTE = 130
   DESCRIPTION = "Bits 7:6 = High Voltage 1.
                       0 = off, 2 = high, 3 = low.
                  Bits 5:4 = High Voltage 2.
                       0 = off, 2 = high, 3 = low.
                  Bits 3:2 = Discriminator level 1.
                       0 = low, 1 = high.
                  Bits 1:0 = Discriminator level 2.
                       0 = low, 1 = high."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = BIN_TIME
   COLUMN_NUMBER = 68
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 131
   DESCRIPTION = "basic binning time for pulses, divide by 2 to
      get microseconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = BIN_SCALING
   COLUMN_NUMBER = 69
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 132
   DESCRIPTION = "Multiplier of how much larger next bin will be."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PULSE_TIME
   COLUMN_NUMBER = 70
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 133
   DESCRIPTION = "Pulsing time in 1/10sec."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = MAX_PULSES
   COLUMN_NUMBER = 71
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 135
   DESCRIPTION = "Maximum number of pulses."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_PROTECTION
   COLUMN_NUMBER = 72
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 137
   DESCRIPTION = "PNG protection options. Least Significant Bit is bit 0.
```

```
                  When bits 0-3 are on, bits 8-15 are ignored
                  bits 11:08 = protection threshold for detector #1.
                  bits 15:12 = protection threshold for detector #2.
                  value 0 = highest, 15 = lowest 15 is lowest protection
                  so most likely to allow PNG_WARNING)
                  bit 3 = '1' to disable warning from PNG HV measurements
                  bit 2 = '1' to disable warning from PNG temperature
sensors
                  bit 1 = '1' to disable warning from detector #2
                  bit 0 = '1' to disable warning from detector #1"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = CTN_SPECTRUM
   COLUMN_NUMBER = 73
   BYTES = 2048
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 139
   ITEMS = 1024
   ITEM_BYTES = 2
   DESCRIPTION = "64 16 channel CTN spectra over time in ACCUM_TIME.
   Epithermal neutrons from 0.4 eV up to energy of about 100 keV."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = CETN_SPECTRUM
   COLUMN_NUMBER = 74
   BYTES = 2048
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 2187
   ITEMS = 1024
   ITEM_BYTES = 2
   DESCRIPTION = "64 16 channel CETN spectra over time in ACCUM_TIME.
   Epithermal and thermal neutrons up to energy of about 100 keV."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_CHECKSUM
   COLUMN_NUMBER = 75
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 4235
   DESCRIPTION = "Checksum at end of instrument data, lowest two bytes
   of the sum of all bytes in data."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FLETCH_CHECKSUM
   COLUMN_NUMBER = 76
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 4237
   DESCRIPTION = "Fletcher checksum added by rover processor."
END_OBJECT = COLUMN
```

5.3 Standby Label

Same keywords as in the Passive Label (Section 5.1), except the column definitions (.FMT file) are as described in Section 5.3.1 below.

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 128
FILE_RECORDS                    = 3

/* POINTERS TO DATA OBJECTS */

^SCIENCE_TABLE                  =
("DNB_417396239EST02240000000_______M1.DAT",1)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-DAN-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS DYNAMIC ALBEDO OF NEUTRONS 2
EDR V1.0"
COMMAND_SEQUENCE_NUMBER         = 0
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = DAN
INSTRUMENT_NAME                 = "DYNAMIC ALBEDO OF NEUTRONS"
INSTRUMENT_TYPE                 = SPECTROMETER
LOCAL_MEAN_SOLAR_TIME           = "Sol-00224M17:22:53:012"
LOCAL_TRUE_SOLAR_TIME           = "16:51:27"
MISSION_NAME                    = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME              = TEST
OBSERVATION_ID                  = UNK
PLANET_DAY_NUMBER               = 224
PRODUCER_INSTITUTION_NAME       = "MULTIMISSION IMAGE PROCESSING
LABORATORY
                                  , JET PROPULSION LAB"
PRODUCT_CREATION_TIME           = 2012-02-03T19:15:49.000
PRODUCT_ID                      = "DNB_417396239EST02240000000_______M1"
PRODUCT_VERSION_ID              = "V1.0 D-22849"
PRODUCT_TYPE                    = DAN_STANDBY
RELEASE_ID                      = "0001"
MSL:REQUEST_ID                  = 0
ROVER_MOTION_COUNTER            = (0,0,0,0,0,0,0,0)
ROVER_MOTION_COUNTER_NAME       = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                  DRILL, RSM, HGA)
SEQUENCE_ID                     = "1036288"
SEQUENCE_VERSION_ID             = "0"
MSL:ACTIVE_FLIGHT_STRING_ID     = B
SOLAR_LONGITUDE                 = -71.8947
SPACECRAFT_CLOCK_CNT_PARTITION  = 1
SPACECRAFT_CLOCK_START_COUNT    = "0417396239.033"
SPACECRAFT_CLOCK_STOP_COUNT     = ""
```

```
START_TIME                        = ""
STOP_TIME                         = ""
TARGET_NAME                       = MARS
TARGET_TYPE                       = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID            = 174
APPLICATION_PROCESS_NAME          = "DAN_STANDBY"
MSL:VIRTUAL_CHANNEL_ID            = 32
MSL:COMMUNICATION_SESSION_ID      = "0"
DOWNLOAD_PRIORITY                 = 31
EARTH_RECEIVED_START_TIME         = 2011-03-23T13:16:02.569
MSL:EXPECTED_TRANSMISSION_PATH    = "3851"
MSL:AUTO_DELETE_FLAG              = "0"
MSL:TRANSMISSION_PATH             = 65535
FLIGHT_SOFTWARE_VERSION_ID        = "97208714"
MSL:FLIGHT_SOFTWARE_MODE          = "8"
MSL:PRODUCT_COMPLETION_STATUS     = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                   = "0"
MSL:SEQUENCE_EXECUTION_COUNT      = 0
RECEIVED_PACKETS                  = 1
SPICE_FILE_NAME                   = "chronos.msl"
TELEMETRY_PROVIDER_ID             = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM     = 18308
MSL:TELEMETRY_SOURCE_SCLK_START   = "417396239.33502"
MSL:TELEMETRY_SOURCE_SIZE         = 384
MSL:TELEMETRY_SOURCE_START_TIME   = 2013-083T11:22:53.327
TELEMETRY_SOURCE_NAME             = "DanStandby_0417396239_51120-1.dat"
MSL:TELEMETRY_SOURCE_TYPE         = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME    = mslatlompcs4
EXPECTED_PACKETS                  = 1

/* SOURCE DATA ELEMENTS */

SOFTWARE_NAME                     = “msledrgen”
SOFTWARE_VERSION_ID               = “1.0”

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

GROUP                             = ROVER_COORDINATE_SYSTEM
 COORDINATE_SYSTEM_NAME           = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX          = (0,0,0,0,0,0,0,0)
 COORDINATE_SYSTEM_INDEX_NAME     = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                    DRILL, RSM, HGA)
 ORIGIN_OFFSET_VECTOR             = (0,0,0)
 ORIGIN_ROTATION_QUATERNION       = (0,0,0,0)
 POSITIVE_AZIMUTH_DIRECTION       = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION     = UP
 REFERENCE_COORD_SYSTEM_NAME      = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX     = 0
END_GROUP                         = ROVER_COORDINATE_SYSTEM

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */
```

```
GROUP                            = RSM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = RSM
 ARTICULATION_DEVICE_NAME        = REMOTE_SENSING_MAST
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = (AZIMUTH, ELEVATION)
END_GROUP                        = RSM_ARTICULATION_STATE

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

GROUP                            = ARM_ARTICULATION_STATE
 ARTICULATION_DEVICE_ID          = ARM
 ARTICULATION_DEVICE_NAME        = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE       = (0<rad>, 0<rad>, 0<rad>, 0<rad>,
0<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME  = ("JOINT 1 SHOULDER AZIMUTH",
                                    "JOINT 2 SHOULDER ELEVATION",
                                    "JOINT 3 ELBOW-ENCODER",
                                    "JOINT 4 WRIST-ENCODER",
                                    "JOINT 5 TURRET-ENCODER")
END_GROUP                        = ARM_ARTICULATION_STATE



/* DATA OBJECT DEFINITION DESCRIPTION */

OBJECT                           = SCIENCE_TABLE
   COLUMNS                       = 63
   INTERCHANGE_FORMAT            = BINARY
   ROW_BYTES                     = 128
   ROWS                          = 3
   DESCRIPTION                   = "
    This table contains neutron spectra and associated instrument
parameters
    as observed by the Mars Science Laboratory (MSL) Dynamic Albedo of
    Neutron Detector (DAN).

    Detailed descriptions for the parameters defined below are
    contained in the DAN EDR SIS document.

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

5.3.1 Standby FORMAT FILE ("DAN_EDR_STDBY.FMT")

```
OBJECT = COLUMN
   NAME = SCLK
   COLUMN_NUMBER = 1
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
   START_BYTE = 1
   DESCRIPTION = "The Rover Compute Element (RCR) clock in seconds."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = AUTOBYTE_CNT
   COLUMN_NUMBER = 2
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 5
   DESCRIPTION = "The number of bytes in the data packet. This includes
FRAME_HDR
   through DAN_CHECKSUM"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FRAME_HDR
   COLUMN_NUMBER = 3
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 9
   DESCRIPTION = "The data frame header. This is broken down into the
following bits:
   bits 1 - 8 is opcode which specifies an instrument activity,
   bits 9 - 10 specifies error control algorithm, DAN uses the checksum
error control,
   bit 11 indicates the presence (1) or absense (0) of both DATA_LEN and
DATA.
   bits 12 - 32 are a status flag defined as follows:
   bit 12 is on if instrument in standby mode,
   bit 13 is on if instrument in passive mode
   bit 14 is on if instrument in active mode
   bit 15 is on indicates PNG warning, an anomily with PNG RCE to turn off
PNG
   bit 16 is on if instrument has taken interesting data
   bits 17 - 32 are unused."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_LEN
   COLUMN_NUMBER = 4
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 13
   DESCRIPTION = "The length of the data packet from the instrument
   in bytes. This includes column SYNCRO through HV_VALUES."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = SYNCRO
   COLUMN_NUMBER = 5
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 17
```

```
   DESCRIPTION = "The synchronization bytes at the beginning of each
record."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DATA_FRAME_NUMBER
   COLUMN_NUMBER = 6
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 21
   DESCRIPTION = "Incrementing counter for data packets."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_TIME
   COLUMN_NUMBER = 7
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 23
   DESCRIPTION = "DAN instrument time in milliseconds since last
                  power up."
END_OBJECT = COLUMN

OBJECT = CONTAINER
   NAME = CMDS_ARRAY
   BYTES = 8
   START_BYTE = 27
   REPETITIONS = 8
   DESCRIPTION = "The last 8 commands sent to the instrument."

   OBJECT = COLUMN
      COLUMN_NUMBER = 8
      NAME = COMMAND_TIME
      BYTES = 4
      START_BYTE = 1
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "DAN instrument time in milliseconds when command
                     was executed."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 9
      NAME = OPCODE
      BYTES = 1
      START_BYTE = 5
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Command identifier."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 10
      NAME = PARAMS
      BYTES = 1
      START_BYTE = 6
      DATA_TYPE = MSB_UNSIGNED_INTEGER
```

```
      DESCRIPTION = "Bit field:
         - bit 7: data present bit, when '0' argument1 and argument2
                  datafields are meaningless
         - bit 6 and 5 are meaningless
         - bits 4 through 0: condition code, shows status of the command
                  execution according to electrical specs."
   END_OBJECT = COLUMN

   OBJECT = COLUMN
      COLUMN_NUMBER = 11
      NAME = ARG1
      BYTES = 1
      START_BYTE = 7
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "First argument to the command."
    END_OBJECT = COLUMN

    OBJECT = COLUMN
      COLUMN_NUMBER = 12
      NAME = ARG2
      BYTES = 1
      START_BYTE = 8
      DATA_TYPE = MSB_UNSIGNED_INTEGER
      DESCRIPTION = "Second argument to the command."
    END_OBJECT = COLUMN

END_OBJECT = CONTAINER

OBJECT = COLUMN
   NAME = FRAME_TYPE
   COLUMN_NUMBER = 48
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 91
   DESCRIPTION = "Current instrument mode, 0 - standby,
                  1 - passive, 2 - active."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP1
   COLUMN_NUMBER = 49
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 92
   DESCRIPTION = "PNG temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP2
   COLUMN_NUMBER = 50
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 93
   DESCRIPTION = "High voltage board temperature in DN."
END_OBJECT = COLUMN
```

```
OBJECT = COLUMN
   NAME = TEMP3
   COLUMN_NUMBER = 51
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 94
   DESCRIPTION = "FPGA temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP4
   COLUMN_NUMBER = 52
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 95
   DESCRIPTION = "Analog board #1 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP5
   COLUMN_NUMBER = 53
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 96
   DESCRIPTION = "Analog board #2 temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = TEMP6
   COLUMN_NUMBER = 54
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 97
   DESCRIPTION = "Case temperature in DN."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_CHARGE_TIME
   COLUMN_NUMBER = 55
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 98
   DESCRIPTION = "PNG charge time in milliseconds"
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_PULSES
   COLUMN_NUMBER = 56
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 99
   DESCRIPTION = "Number of PNG pulses since power on."
END_OBJECT = COLUMN
```

```
OBJECT = COLUMN
   NAME = NUM_CMD_RECV
   COLUMN_NUMBER = 57
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 101
   DESCRIPTION = "Number of commands received since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = NUM_CMD_RJCT
   COLUMN_NUMBER = 58
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 103
   DESCRIPTION = "Number of commands rejected since power on."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = PNG_ALARM
   COLUMN_NUMBER = 59
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 105
   DESCRIPTION = "Indication of PNG Alarm, bifield:
                  Bit 7: Alarm by PNG HV out of range
                  Bit 6: PNG temperature out of range
                  Bit 5: Detector 2 anomaly countrate
                  Bit 4: Detector 1 anomaly countrate
                  Bit 0: PNG alarm detected."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = RESET_TYPE
   COLUMN_NUMBER = 60
   BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 106
   DESCRIPTION = "Type of reset:
                  0 - Power on reset
                  1 - RS-422 cmd reset
                  2 - Discrete line A reset
                  3 - Discrete line B reset."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = HV_VALUES
   COLUMN_NUMBER = 61
   BYTES = 16
   ITEMS = 16
   ITEM_BYTES = 1
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 107
   DESCRIPTION = "History of the PNG high voltage, last 16 values.
   Real value in Volts calculated by 1 / (17 * value / 1600000)"
```

```
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = DAN_CHECKSUM
   COLUMN_NUMBER = 62
   BYTES = 2
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 123
   DESCRIPTION = "Checksum at end of instrument data, lowest two bytes
   of the sum of all bytes in data."
END_OBJECT = COLUMN

OBJECT = COLUMN
   NAME = FLETCH_CHECKSUM
   COLUMN_NUMBER = 63
   BYTES = 4
   DATA_TYPE = MSB_UNSIGNED_INTEGER
   START_BYTE = 125
   DESCRIPTION = "Fletcher checksum added by rover processor."
END_OBJECT = COLUMN
```

6. APPENDIX A - DAN LABEL KEYWORD DEFINITIONS

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
PDS_VERSION_ID	Specifies the version number of the PDS standards document that is valid when a data product label is created. Values for the PDS_version_id are formed by appending the integer for the latest version number to the letters 'PDS'.	String		PDS3	
/* FILE DATA ELEMENTS */	Comment				
RECORD_TYPE	A Specifies the record format of a file. Note: In the PDS, when RECORD_TYPE is used in a detached label file, it always describes its corresponding detached data file and 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		FIXED_LENGTH	Constant
RECORD_BYTES	B Specifies the number of bytes in a physical file record, including record terminators and separators. Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.	integer		2048	Product specific

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
FILE_RECORDS	Specifies the number of physical file records, including both label records and data records. Note: In the PDS the use of FILE_RECORDS along with other file-related data elements is fully described in the Standards Reference.	integer		1038	Calculated
/* IDENTIFICATION DATA ELEMENTS */	Comment				
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)		See example labels for actual values used in each product type	Constant for each instrument.
DATA_SET_NAME	Specifies the full name given to a data set or a data product. The DATA_SET_NAME typically identifies the instrument that acquired the data, the target of that instrument, and the processing level of the data. In the PDS, values for DATA_SET_NAME are constructed according to standards outlined in the Standards Reference.	String			PDS Table lookup

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
COMMAND_SEQUENCE_NUMBER	Specifies a numeric identifier for a sequence of commands sent to a spacecraft or instrument. Note: For MSL, this is the command number that identifies the specific generating command within the specified sequence.	String		33	EMD: CommandNumber
INSTRUMENT_HOST_ID	Specifies 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		"MSL"	Scid
INSTRUMENT_HOST_NAME	The full name of the host on which this instrument is based	String		"MARS SCIENCE LABORATORY"	Scid
INSTRUMENT_ID	Specifies an abbreviated name or acronym which identifies an instrument	String		"DAN"	
INSTRUMENT_NAME	Name of the instrument, free format, enclosed in double quotes. See example labels for various names used for MECA non-imaging products	String		"DYNAMIC ALBEDO OF NEUTRONS"	EMD: ProductName
INSTRUMENT_TYPE	Specifies the type of an instrument.	String		"SPECTROMETER"	

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
LOCAL_MEAN_SOLAR_TIME	Specifies the local mean solar time, or LMST. It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. The desire to work with solar days, hours, minutes, and seconds of uniform length led to the concept of the fictitious mean Sun or FMS. The FMS is defined as a point that moves on the celestial equator of a planetary body at a constant rate that represents the average mean motion of the Sun over a planetary year. Local mean solar time is defined, by analogy with local true solar time (LTST), as the difference between the areocentric right ascensions of a point on the surface and of the FMS. The difference between LTST and LMST varies over time. The length of a mean solar day is constant and can be computed from the mean motion of the FMS and the rotation rate of a planet. The mean solar day is also called a 'sol'. Mean solar hours, minutes, and seconds are defined in the same way as the true solar units. For MSL, the valid value is expressed in terms of a 24-hour clock, so the acceptable range is “00:00:00.000” to “23:59:59.999”. See also LOCAL_TRUE_SOLAR_TIME.	String		Sol-<nnnnn>M<hh>:<mm>:<ss>[.fff] NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	Calculated: - SCLK kernel - Landing site kernel - P kernel

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
LOCAL_TRUE_SOLAR_TIME	Specifies the local true solar time, or LTST. It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and areocentric 'right ascension' and 'declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet's surface. Specifically, LTST = (a(P) - a(TS)) * (24 / 360) + 12 where, LTST = the local true solar time in	String		hh:mm:ss[.fff] NOTE: Value will be uncalibrated if SPICE kernels are unavailable.	Calculated: - SCLK kernel - Landing site kernel - P kernel

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
PLANET_DAY_NUMBER	Specifies the number of sidereal days (rotation of 360 degrees) elapsed since a reference day (e.g., the day on which a landing vehicle set down). Days are measured in rotations of the planet in question from the reference day. For MSL, the reference day is "1", as Landing day is Sol 1. If before Landing day, then value will be less than "1" and can be negetive.	Integer			Calculation - SCLK kernel
MISSION_NAME	Specifies a major planetary mission or project. A given planetary mission may be associated with one or more spacecraft.	String		"MARS SCIENCE LABORATORY"	Static Value
MISSION_PHASE_NAME	Specifies the commonly-used identifier of a mission phase.	String		"DEVELOPMENT", "LAUNCH", "CRUISE AND APPROACH", "ENTRY, DESCENT, AND LANDING", "PRIMARY SURFACE MISSION", "EXTENDED SURFACE MISSION"	User specified parameter.
OBSERVATION_ID	Specifies a unique identifier for a scientific observation within a data set. It is set via the data product context ID - which doesn't necessarily map to a specific object - it's just used to group various instrument data sets together via a common keyword.	String		UNK	Static value

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
PRODUCER_INSTITUTION_NAME	Specifies the identity of a university, research center, NASA center or other institution associated with the production of a data set. This would generally be an institution associated with the element PRODUCER_FULL_NAME.	String		"MULTIMISSION INSTRUMENT PROCESSING LABORATORY, JET PROPULSION LAB"	Static Value.
PRODUCT_CREATION_TIME	Defines the UTC system format time when a product was created.	String		YYYY-MM-DDThh:mm:ss.fff	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)		File name, less the extension.	Filename minus the extension
PRODUCT_VERSION_ID	Specifies the version of an individual product within a data set. PRODUCT_VERSION_ID is intended for use within AMMOS to identify separate iterations of a given product, which will also have a unique FILE_NAME.	String		"V<*vernum*> D-38107"	User specified parameter

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
RELEASE_ID	Specifies the unique identifier associated with the release to the public of all or part of a data set. The release number is associated with the data set, not the mission. When a data set is released incrementally, such as every three months during a mission, the RELEASE_ID is updated each time part of the data set is released. The first release of a data set in the mission should have a value of "0001". For example, on MSL the first release of the SSI EDR data set on MSL will have RELEASE_ID = "0001". The next SSI EDR release will have RELEASE_ID = "0002".	String			User parameter input.
REQUEST_ID	Specifies the Request ID value associated with the Data Product generation command. Unsigned integer.				EMD: RequestId

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ROVER_MOTION_COUNTER	Specifies a set of integers which describe a (potentially) unique location (position/orientation) for a rover. Each time something happens that moves, or could potentially move, the rover, a new motion counter value is created. This includes intentional motion due to drive commands, as well as potential motion due to other articulating devices, such as arms or antennae. This motion counter (or part of it) is used as a reference to define instances of coordinate systems which can move such as SITE or ROVER frames. The motion counter is defined in a mission-specific manner. Although the original intent was to have incrementing indices (e.g. MER), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values. For MSL, the motion counter consists of eight values. In order, they are Site, Drive, Pose, Arm, Chimra, Drill, RSM 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.			(1, 2, 0, 0, 0, 0, 0, 0)	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, Chimralndex, DrillIndex, RsmIndex, HgaIndex)
ROVER_MONTION_COUNTER_NAME	Specifies an array that provides the formal names identifiying each integer in ROVER_MOTION_COUNTER.			(SITE, DRIVE, POSE, ARM, CHIMRA, DRILL, RSM, HGA)	Constant

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
SEQUENCE_ID	Specifies an identification of the spacecraft sequence associated with the given product. This element replaces the older seq_id, which should no longer be used.			P0171	EMD: SequenceId
SEQUENCE_VERSION_ID	Specifies the version identifier for a particular observation sequence used during planning or data processing.			2	EMD: SequenceVersion
SPACECRAFT_CLOCK_CNT_PARTITION	Specifies the clock partition active for SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT elements.	Integer		"1"	Static value
SPACECRAFT_CLOCK_START_COUNT	Starting SCLK, smallest, value of all the records contained in the EDR.	string(30)		Format is ddddddddd.ddd, measured in units of seconds stored internally as a floating point number.	- EMD: DvtCourse/DvtFine or - Sclk or - Pulled from instrument data
START_TIME	SPACECRAFT_CLOCK_START_COUNT converted and represented in UTC	string		Formation rule: YYYY-MM-DDThh:mm:ss.fff	OnBoardCreationTime – the coarse SCLK in 1-second presentation
TARGET_NAME	Identifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE. This is based on mission phase.	string(30)		"MARS", "CALIBRATION"	Calculated by algorithm to determine if looking at the calibration target, if not, then MARS.
TARGET_TYPE	Specifies the type of a named target.	string		CALIBRATION, DUST, N/A, SUN, PLANET	Static value
/* TELEMETRY DATA ELEMENTS */	Comment				
APPLICATION_PROCESS_ID	Identifies the source/process which created the data.	Integer			EMD: ApId

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
APPLICATION_PROCESS_NAME	Provides the name associated with the source/process which created the data.	String(256)			EMD: ProductName
VIRTUAL_CHANNEL_ID	The Virtual Channel Identifier is used by MSL to identify the RCE string generating the Transfer Frame, and to indicate the type of data flowing in the telemetry virtual channel. RCE String A is indicated by all Virtual Channel Identifier values having a '0' as the high bit (e.g., virtual channels 0 to 31); RCE String B is indicated by all Virtual Channel Identifier values having a '1' for the high bit (e.g., virtual channels 32 to 63).			NULL	EMD: VcId
COMMUNICATION_SESSION_ID	Active Communication Session ID at time of MPDU creation.				EMD: CommSessionId
DOWNLOAD_PRIORITY	Specifies which data to downlink/transmit, based on order of importance. The lower numerical priority (higher-ranked number) data products are transmitted before higher numerical priority (lower-ranked number) data products. For example, an image with a downlink priority of 25 will be transmitted from the rover before an image with a downlink priority of 50.			0	EMD: ProductPriority
EARTH_RECEIVED_START_TIME	Earth Receive Time (ERT)	String		Formation rule: YYYY-MM-DDThh:mm:ss.fff	EMD: GroundCreationTime, or FirstPartErt
EXPECTED_TRANSIMISSION_PATH	Routing control at time of MPDU generation. Indicates the planned transmission paths (routes) for the Data Product			NULL	EMD: TransmissionControlCriterion

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
AUTO_DELETE_FLAG	Indicates if the DP will be deleted upon transmission: 1 for delete, 0 for no delete.			NULL	EMD: DeleteOnSend
TRANSMISSION_PATH	Routing status at time of MPDU generation. Indicates the actual transmission paths (routes) of the Data Product.			NULL	EMD: TransmissionStatus
FLIGHT_SOFTWARE_VERSION_ID	Identifies the flight software version	String			EMD: FswVersion
FLIGHT_SOFTWARE_MODE	Active Flight Software version at Data Product creation				FswMode
PRODUCT_COMPLETION_STATUS	Identifies the completion level of the product	String		"COMPLETE" "PARTIAL"	EMD: GroundStatus
PRODUCT_TAG	Data Product Tag. Comment: Use of this tag is defined separately for individual product types. It is anticipated that this tag may be used to associate multiple products for later processing; it may also be used to indicate instrument FSW versions, or other uses	String		N/A	EMD: ProductTag
SEQUENCE_EXECUTION_CNT	Set to 0 at RCE start-up and Incremented each time this sequence has executed since last RCE start-up. Unsigned integer.	String		N/A	EMD: SequenceExecutionCounter
ACTIVE_FLIGHT_STRING_ID	Set to 0 at RCE start-up and Incremented each time this sequence has executed since last RCE start-up. Unsigned integer				EMD: CreationStringId
TELEMETRY_PROVIDER_ID		String		MPCS_MSL_DP	Constant
TELEMETRY_SOURCE_SCLK_START	Spacecraft Time in seconds since epoch, DVT. Coarse portion of the data validity time, from the secondary packet header.				EMD: DvtCoarse/DvtFine
TELEMETRY_SOURCE_START_TIME	A SCET calculated by MPCS using the Data Validity time and the SCLK/SCET correlation file.				EMD: Scet

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
TELEMETRY_SOURCE_SIZE	Indicates the number of bytes in a physical file record, including record terminators and separators.	Integer		0 to n	EMD: ProductFileSize
TELEMETRY_SOURCE_CHECKSUM	File checksum is an unsigned add of each byte in the data areas of the DPOs in the product. This does not include the DPO headers.				EMD: ProductChecksum
RECEIVED_PACKETS	Specifies the total number of telemetry packets that constitute a reconstructed data product. For MSL, "Packets" are also referred to as "Parts".			0 to n	EMD: PartList:TotalReceived
SPICE_FILE_NAME	Specifies the names of the SPICE files used in processing the data. For Galileo, the SPICE files are used to determine navigation and lighting information.	String			User parameter input
TELEMETRY_SOURCE_NAME	Identifies the telemetry originator	String		340477575DNA_0000EPA00_____J1.DAT	Data File Name
TELEMETRY_SOURCE_TYPE		String		"DATA PRODUCT"	Constant
TELEMETRY_SOURCE_HOST_NAME	Identifies the VENUE HOST name	String			EMD: Venue:Host
EXPECTED_PACKETS					EMD: TotalParts
/* SOURCE DATA ELEMENTS */	Comment				
SOFTWARE_NAME	Identifies data processing Software, such as a program or a program library that was used to generate the EDR.	String		"msledrgen"	Constant
SOFTWARE_VERSION_ID	Indicates the version (development level) of a program or a program library.	String		"1.0"	User parameter input
/* COORDINATE SYSTEM STATE: ROVER */	Comment				

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
COORDINATE_SYSTEM_NAME		String		ROVER_FRAME	Constant
COORDINATE_SYSTEM_INDEX	Specifies a set of integers which describe a (potentially) unique location (position/orientation) for a rover. Each time something happens that moves, or could potentially move, the rover, a new motion counter value is created. This includes intentional motion due to drive commands, as well as potential motion due to other articulating devices, such as arms or antennae. This motion counter (or part of it) is used as a reference to define instances of coordinate systems which can move such as SITE or ROVER frames. The motion counter is defined in a mission-specific manner. Although the original intent was to have incrementing indices (e.g. MER), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values. For MSL, the motion counter consists of eight values. In order, they are Site, Drive, Pose, Arm, Chimra, Drill, RSM 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.	String(12)		(1,2,0,0,0,0,0,0)	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, ChimraIndex, DrillIndex, RsmIndex, HgaIndex)
COORDINATE_SYSTEM_INDEX_NAME	Specifies an array that provides the formal names identifying each integer in COORDINATE_SYSTEM_INDEX.	string		(SITE, DRIVE, POSE, ARM, CHIMRA, DRILL, RSM, HGA)	RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, ChimraIndex, DrillIndex, RsmIndex, HgaIndex)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ORIGIN_OFFSET_VECTOR	Specifies the offset from the reference coordinate system's origin to the origin of the coordinate system being defined by the enclosing COORDINATE_SYSTEM group. In other words, it is the location of the current system's origin as measured in the reference system.	string(30)		(0.0230152, -0.076101, 0.874005)	EMD: RoverPosition (PositionX, PositionY, PositionZ)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ORIGIN_ROTATION_QUATERNION	Specifies an array of four values that specifies the rotation of the coordinate system being defined by the enclosing COORDINATE_SYSTEM group, relative to the reference system. Mathematically this can be expressed as follows: Given a vector expressed in the current frame, multiplication by this quaternion will give the same vector as expressed in the reference frame. Quaternions are expressed as a set of four numbers in the order: (s, v1, v2, v3) where, s = cos(theta/2) v(n) = sin(theta/2)*a(n). theta = the angle of rotation a = (x,y,z) vector around which the rotation occurs. For MSL, the value for ORIGIN_ROTATION_QUATERNION that defines a coordinate frame like Rover frame is computed with respect to only the orientations of the frame's axes… regardless of whether POSITIVE_ELEVATION_DIRECTION is declared to be “UP” or “DOWN”.	String(60)		(0.922297, -0.0165226, -0.0413094, 0.382304)	EMD: RoverAttitude (AttitudeX, AttitudeY, AttitudeZ, AttitudeW)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
POSITIVE_AZIMUTH_DIRECTION	Specifies the direction in which azimuth is measured in positive degrees for an observer on the surface of a body. The azimuth is measured with respect to the elevational reference plane. A value of CW indicates that Azimuth is measured positively Clockwise, and CCW indicates that Azimuth increases positively Counter-clockwise	String(20)		CLOCKWISE	static
POSITIVE_ELEVATION_DIRECTION	Specifies the direction in which elevation is measured in positive degrees for an observer on the surface of a body. The elevation is measured with respect to the azimuthal reference plane. A value of “UP” indicates that elevation is measured positively upwards, i.e., the zenith point would be at +90 degrees and the nadir point at -90 degrees. “DOWN” indicates that the elevation is measured positively downwards; the zenith point would be at -90 degrees and the nadir point at +90 degrees	String		UP	static

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
REFERENCE_COORDINATE_SYSTEM_NAME	Specifies 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		SITE_FRAME	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, ChimraIndex, DrillIndex, RsmIndex, HgaIndex)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
REFERENCE_COORDINATE_SYSTEM_INDEX	Specifies which instance of the coordinate system named by REFERENCE_COORD_SYSTEM_NAME is the reference coordinate system for the group in which the keyword occurs. This index is a set of integers which serve to identify coordinate system instances in a mission-specific manner. 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		1	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, ChimraIndex, DrillIndex, RsmIndex, HgaIndex)
/* ARTICULATION_DEVICE_STATE: REMOTE SENSING MAST */	Comment				

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ARITCULATION_DEVICE_ID	Specifies the unique abbreviated identification of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g., mast heads, wheel bogies, arms, etc.). Note: The ARTICULATION_DEVICE_ID is not a unique identifier for a given articulated device. Note also that the associated ARTICULATION_DEVICE_NAME element provides the full name of the articulated device.	string(30)		RSM_ARITCULATION_STATE	static
ARTICULATION_DEVICE_NAME	Specifies the common name of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g. mast heads, wheel bogies, arms, etc.) Note: The associated ARTICULATION_DEVICE_ID element provides an abbreviated name or acronym for the articulated device.	string		REMOTE_SENSING_MAST	static
ARTICULATION_DEVICE_ANGLE	Specifies the value of an angle between two parts or segments of an articulated device. Note: MSL uses radians. The PDS default unit for this keyword is degrees, so the <rad> tag is required for MSL data.	string		(0.0230152 <rad>, -0.076101 <rad>	EMD: RemoteSensingMastOrientation (Azimuth, Elevation)
ARTICULATION_DEVICE_ANGLE_NAME	Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE.	string			EMD: RemoteSensingMastOrientation (Azimuth, Elevation)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
/* ARTICULATION DEVICE STATE: ROBOTIC ARM DEVICE AT DP CREATION */	Comment				
ARTICULATION_DEVICE_ID	Specifies the unique abbreviated identification of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g., mast heads, wheel bogies, arms, etc.). Note: The ARTICULATION_DEVICE_ID is not a unique identifier for a given articulated device. Note also that the associated ARTICULATION_DEVICE_NAME element provides the full name of the articulated device.	string		ARM	static
ARTICULATION_DEVICE_NAME	Specifies the common name of an articulation device. An articulation device is anything that can move independently of the spacecraft to which it is attached, (e.g. mast heads, wheel bogies, arms, etc.) Note: The associated ARTICULATION_DEVICE_ID element provides an abbreviated name or acronym for the articulated device.	String		ROBOTIC_ARM	static

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ARTICULATION_DEVICE_ANGLE	Specifies the value of an angle between two parts or segments of an articulated device. Note: MSL uses radians. The PDS default unit for this keyword is degrees, so the <rad> tag is required for MSL data.	string		(0.0230152 <rad>, -0.076101 <rad>, -0.076101 <rad>, -0.076101 <rad>, -0.076101 <rad>)	EMD: SampleArmOrientation (JointAngle1, JointAngle2, JointAngle3, JointAngle4, JointAngle5)
ARITCULATION_DEVICE_ANGLE_NAME	Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE.	Integer		(“JOINT 1 SHOULDER AZIMUTH”, “JOINT 2 SHOULDER ELEVATION”, “JOINT 3 ELBOW-ENCODER”, “JOINT 4 WRIST-ENCODER”, “JOINT 5 TURRET-ENCODER”)	EMD: SampleArmOrientation (JointAngle1, JointAngle2, JointAngle3, JointAngle4, JointAngle5)